

A GENERAL RECTILINEAR-DISTANCE
LOCATION-ALLOCATION PROBLEM

A THESIS

Presented to

The Faculty of the Division of Graduate Studies

By

Hanif D. Sherali


In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in Operations Research

Georgia Institute of Technology


December 1976

A GENERAL RECTILINEAR-DISTANCE
LOCATION-ALLOCATION PROBLEM


Approved:



Dr. C. M. Shetty, Chairman



Dr. L. F. McGinnis



Dr. D. E. Fyffe

Date approved by Chairman Nov. 5th 1976.

ACKNOWLEDGMENTS

It gives me joy to avail of this opportunity to show my deep appreciation for and express my most sincere and heartfelt gratitude to Dr. C. M. Shetty for all the assistance rendered to me during the development of this dissertation. His constant guidance and encouragement and his contagious enthusiasm have been the most influential factors in my achievements at the Georgia Institute of Technology.

I wish to thank Dr. L. F. McGinnis and Dr. D. E. Fyffe for serving on my Thesis Advisory Committee and for providing helpful criticism. I also wish to thank Dr. R. Heikes for his useful comments on the design and analyses of the computational experience.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	ii
LIST OF TABLES	v
SUMMARY	vi
Chapter	
I. INTRODUCTION	1
Introduction	
Problem Statement	
Literature Survey of Related Problems	
Summary of Proposed Approach	
II. BASIC PROPERTIES OF THE GENERAL RECTILINEAR-DISTANCE LOCATION-ALLOCATION PROBLEM (GRLAP)	14
Introduction	
Transformation of GRLAP into BLP	
Fundamental Definitions and Theorems	
Investigation of Some Basic Properties of Problems P_1 and P_2	
The Allocation and P_2 Location Aspects of Problem GRLAP1	
III. THE INTERACTING MULTIFACILITY LOCATION PROBLEM	26
Introduction	
Problem Reformulation and Description	
Consideration of Single Movements	
Consideration of Joint Movements	
A Complete Algorithm (MFLOC) For the Solution of Problem P_x	
Illustrative Examples	
IV. THE CUTTING PLANE AND FEASIBLE POINT ALGORITHMS	
Introduction	
The Cutting Plane Algorithm for Bilinear Programming Problems	
The Negative Edge Extension Method	
Cardinality of Set J	
Development of \bar{e}^j and Determination of $\psi(\lambda_j)$ and $\bar{\psi}(\lambda_j)$	

TABLE OF CONTENTS (Continued)

	Page
Solution of Problems PAR 1 and PAR 2	
Determination of a Weak Pseudo Global Minimum	
Determination of a Good Starting Solution	
Determination of a Feasible Extreme Point to the Set of Cuts	
Algorithm "Feas"	
V. THE ALGORITHM TO SOLVE A GENERAL RECTILINEAR-DISTANCE LOCATION-ALLOCATION PROBLEM	91
Statement of the Algorithm	
Illustrative Examples	
VI. COMPUTATIONAL RESULTS, SUMMARY AND CONCLUSIONS	104
Introduction	
Computational Results	
Summary of Specific Contributions	
Conclusions and Recommended Research	
BIBLIOGRAPHY	115

LIST OF TABLES

Table	Page
1. Computational Experience with the Revised Pritsker-Ghare (PG) Facility Locating Procedure and with the MFLOC Procedure Presented in this Study (H)	105
2. Newton's Search vs. Bolzano's Search Technique	106
3. Comparison of Computational Experience with The Solution Procedures of Morris, Sherali A., and the Proposed Procedure	107
4. Computational Times for Problem GRLAP	108
5. 3^3 Factorial Analyses for Linear and Quadratic Effects	109
6. First Times to Best Recorded Solutions (K=1; No Interactions Between New Facilities)	111
7. First Times to Best Recorded Solution (K=5; With Interactions Between New Facilities)	111

SUMMARY

In this thesis, an efficient exact solution procedure is developed to solve a General Rectilinear Distance Location-Allocation Problem. This problem involves a specified number of existing facilities having a known deterministic demand for several products. A specified number of new facilities, having a known capacity to supply these products at given cost values, are to be located. The new facilities may also have known demands for the products. The objective is to locate these sources so as to minimize the cost of purchase of the products and of their transportation (using rectilinear distance measure).

Basically, a cutting plane solution technique is employed. The procedure initiates from a good starting solution and works towards a pseudo-global minimum. (As a sub-problem in this step, a solution to the interacting multifacility location problem is provided.) A deep cut is then introduced to eliminate as much of the non-improving feasible region as possible. Thereafter, at each stage, an extreme point solution is determined which is feasible to the system of cuts introduced. Starting from this point, the procedure works towards what is called a "weak pseudo-global minimum." Another cut is now introduced and the procedure continues until the entire feasible region is exhausted. The optimum solution is the best among the first pseudoglobal minimum and the subsequent weak pseudo-global minima obtained.

Computational experience is provided on a CDC Cyber 74 computer. The procedure and some of its subproblems are tested against known solution

procedures. Problems which can be solved within fifteen minutes of computational time are reported. An analysis of variance is conducted to study the sensitivity of computational time to the number of facilities (new and existing) and the number of products. A second order regression model is built to serve as an approximate prediction equation for the computational time.

CHAPTER I

INTRODUCTION AND LITERATURE SURVEY

1.1 Introduction

Several economic ventures involve location and allocation of scarce resources. Before the advent of formalized techniques of Operations Research, one had to depend on experience and intuition to handle these problems. Today, there exists an ever increasing number of quantitative approaches to assist in solving location and allocation problems.

Kuhn (23) relates that in the seventeenth century, Fermat formulated what is believed to be the first formal facility location problem. "Given three points in a plane, find a fourth point such that the sum of its distances to the three given points is a minimum." A simple geometric solution to this was provided by Toricelli before 1640 (41). This problem was later modified to consider more than three existing facilities and came about to be referred to as the Steiner-Weber problem. However, it was not until 1962 that Kuhn and Kuenne (24) provided a satisfactory iterative solution technique for this problem. Since then, several contributions to literature have dealt with variations and extensions of the Steiner-Weber problem. We will first introduce our problem and then briefly discuss other related problems of interest to us.

1.2 Problem Statement

The problem we deal with here is a general rectilinear distance

location-allocation problem having several products, sources and destinations. It involves interactions between sources (new facilities to be located) and between sources and destinations (existing facilities). Let there be m existing facilities and suppose that we are to locate n new facilities. Let the new facilities be numbered 1 through n and let the existing facilities be numbered $(n+1)$ through $(n+m)$. Let the new facility i have a capacity a_{ik} to manufacture product k ($k=1, \dots, K$). Let us assume that the source or destination i has a demand b_{ik} for product k . Let the cost of a unit product k when it comes from the i^{th} source be c_{ik} and let t_k be the cost of transporting it through a unit distance. We are then to determine the location of the new facilities and the allocations u_{ijk} of product k from facility i to facility j so as to minimize the total cost of purchase and of transportation (using rectilinear distance measure).

To state the problem mathematically, let (x_i, y_i) denote the location of the i^{th} facility relative to some coordinate system. $((x_i, y_i))$ are decision variables for $i=1, \dots, n$ and are fixed and equal to (d_i, e_i) for each facility $(n+i)$, $i=1, \dots, m$. Let us denote the rectilinear distance between facilities i and j by

$$d(i, j) = |x_i - x_j| + |y_i - y_j|.$$

We may then formulate the problem as:

PROBLEM GRLAP:

$$\min \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^{n+m} [c_{ik} + t_k d(i, j)] u_{ijk}$$

subject to

$$\sum_{j=1}^{n+m} u_{ijk} \leq a_{ik} \quad i = 1, \dots, n; \quad k = 1, \dots, K.$$

$$\sum_{i=1}^n u_{ijk} = b_{jk} \quad j = 1, \dots, n+m; \quad k = 1, \dots, K.$$

$$u_{ijk} \geq 0 \quad \forall i, j, k.$$

Our main objectives of this study are:

1. To develop an efficient solution procedure for the above problem. For this purpose, we will also develop an efficient procedure for a subproblem, namely, the pure location problem.

2. To test the computational efficiency of our procedure against known solution procedures to closely related special cases of the problem, and also to determine the size of the problem that can be solved in less than fifteen minutes of computer time.

A more detailed statement of the objectives is given in Section 1.4.

It may be noted (and we will show this in Chapter II), that our problem may be cast into the general framework of a Bilinear Programming Problem. Vaish (41) gives a good account of several generalized techniques available to solve this class of problems. We will appreciably modify one such technique and develop an exact solution procedure for our problem.

1.3 Literature Survey of Related Problems

In this section, we will first discuss some solution procedures

for the pure location problem i.e. where the allocations are specified. We will classify this problem according to the solution space characteristics and discuss in greater detail the problem involving location in a plane. We will subclassify the latter according to the mode of distance measure adopted. We will not discuss its counterpart, the pure allocation problem and we refer the reader to reference (2) for this. We will then review location-allocation problems. There again, we will categorize the solution techniques according to the distance measure adopted.

1.3.1 Pure Location Problems

1. Location on a Network. In contexts similar to the location of Industrial Plants, we are required to profitably locate new facilities on a subset of a finite number of potential sites. Such problems are referred to as discrete plant location problems or Network location problems. Effroymsen and Ray (12) formulate the discrete plant location problem as a mixed integer problem. Initially, they ignore the integer restrictions and solve the problem of location in a plane using linear programming techniques. If the resulting solution is not integral, a branch and bound approach is adopted to obtain an integer solution. A variety of plant location problems have been treated by both heuristic and exact procedures. References (19,21,36) relate to some of the research done in this area.

2. The Covering Problem. Another plant location problem which is frequently encountered is the covering problem. There, a specification of the maximum permissible distance between existing and new facilities

is made. The decision variables are the number of new facilities and their location. Such problems arise when locating public utility facilities like schools, police stations, health-care centers, post-offices, etc. Generally speaking, there are four major approaches reported in the literature for solving covering problems. One of these is an implicit enumeration approach such as the branch and bound technique (25). Another approach uses cutting planes and solves iteratively, a number of linear programming problems (3). A third approach employs reduction techniques (40), and a fourth deals with heuristic methods (18).

3. Location on a Plane. Problems involving a continuous solution space fall into this category. Common examples are facility layout and warehouse design and location of communication networks. We will classify these problems according to the mode of distance measure adopted.

(a) Rectilinear Distance Location Problems. In the context of location in a grid of city streets or in a network of aisles in a factory or a warehouse (14), rectilinear distance measure gives the best approximation to the actual situation. An attempt to give an exact solution procedure to the most general form of this problem i.e. one involving several new and existing facilities with interactions between new facilities and between new and existing facilities was made by Pritsker and Ghare (34). They formulated a perturbation model for this problem. Using a primal simplex based approach, Rao (35) verified this procedure for the non-degenerate case. He then gave a counter example to show that the optimality conditions proposed by Pritsker and Ghare were not sufficient if degeneracy existed. Pritsker (33) then presented corrective

steps but, however, specified conditions under which optimality may still not be claimed. In Chapter III we will develop an exact primal simplex based algorithm to solve this problem. In Chapter VI we will report computational times and compare them with those reported by Pritsker (33).

It has often been conjectured (44) that a dual solution to this problem is more amenable to standard solution techniques. Cabot et al. (8) have aptly demonstrated this through their network flow solution procedure. However, they remark that if certain constraints are added which restrict the location of new facilities to only some specified points in the solution space, the procedure is no longer applicable. Unfortunately, computational times have not been reported here.

Another type of approach to this problem was developed by Eyster et al. (13). They presented a hyperboloid approximation procedure (HAP) which approximated the objective function at the points where its derivative is not defined. Use was then made of a gradient procedure to yield a solution which was subsequently improved by fixed point iteration methods. At each stage, the approximated function was improved until a suitable stopping criteria was met with. The approximation used was $|x_i - d_j| = [(x_i - d_j)^2 + \epsilon]^{1/2}$ where ϵ tends to zero. However, no proof of convergence was presented.

The problem which involves no interaction between new facilities is separable with respect to the new facilities and the solution procedure is simple and straightforward (44). A primal simplex based proof for this is presented in reference (37). A stronger result for this

problem is provided by Morris and Love (98), who show how the feasible solution space may be reduced.

An interesting modification of this problem is presented by Wesolowsky and Love (43). Their formulation permits the destinations to be single points, lines or rectangular areas. A gradient reduction solution procedure is described which has the property that the direction of descent is determined by the geometric properties of the problem. Since points and overlapping rectangular areas can be used to approximate quite complex spatial distributions, this method would seem to have applicability in urban location problems like those concerning postal districts (4) and some forms of facility design (15).

(b) Euclidean Distance Location Problems. In this class of problems, the distance measure $d(i,j)$ is given by $d(i,j) = [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2}$. About the best available algorithm to solve this problem has been provided by Eyster et al. (13). They use a Hyperboloid Approximation Procedure very similar to that used for the rectilinear distance problem. The approximation used gives rise to the expression $d(i,j) = [(x_i - x_j)^2 + (y_i - y_j)^2 + \epsilon]^{1/2}$ where ϵ tends to zero. Hence, the derivative of $d(i,j)$ is defined for every (x_i, y_i) in E_2 .

Another procedure is the modification of this due to Kuhn (23), based on the definition of the derivative of $d(i,j)$ in the feasible region. This procedure also uses fixed point iteration methods to obtain the optimum locations. However, unlike the HAP procedure, a proof of convergence is available here. A more detailed discussion is given by Kuhn and Kuenne (24).

Some heuristic procedures in this field are due to Cooper (11). He writes out the objective function using zero-one variables to identify whether a new facility (n in number) services an existing facility (m in number). He then takes partial derivatives of this expression with respect to the location decision variables and equates them to zero. The resulting equations are solved using some rapidly converging iterative techniques. The heuristic Procedures are:

(i) Destination Subset Algorithm. Here it is claimed that if one considered all possible subsets of n destinations from the m total destinations to locate the sources, a close approximation to the optimal solution is obtained. The reader may refer to Lehmer (26) for an excellent method of generating mC_n on a digital computer. However, even with careful coding, when a run was made on an IBM 7072 machine with $n=4$ and $m=60$, it took 3.5 hours to arrive at the optimal solution.

(ii) Random Destination Algorithm. This gives reasonable computation times. Here again, the destination set is assumed best for locating new facilities. However, location is done by a random generator till an aspired limit is obtained.

(iii) Successive Approximation Algorithm. Here, two sources are fixed as in (i) above. Then, the third new facility is placed at each of the n destinations and the one which yields the best objective function value is selected for locating this new facility. The remaining sources are located similarly.

(c) Euclidean Distance Squared Location Problems. This form of distance measure is a convenient approximation of the Euclidean Distance measure. Kuhn and Kuenne (24) show that this problem may be solved

through a mechanical analogue. This amounts to fixing at each existing facility a weight proportional to its interaction with the new facility being located. The center of gravity of this system of weights gives the optimum location of that new facility.

Some reported applications are in plant locations and evaluation of alternate factory sites (5) and in communication networks.

(d) Generalized Distance Location Problems. Location problems have also been solved using generalized distance measures of which rectilinear and Euclidean distances are special cases. Wesolowsky (42) has presented location models solved under minimax optimization criteria. He uses parametric programming as a modification of the geometric techniques of minimizing the maximum distance of a source to the destinations.

Morris and Love (27) have also some results in this area. They provide a solution for a constrained multi-facility location problem involving ℓ_p distances using convex programming. The ℓ_p distance in E_n is defined to be $\ell_p(Q,R) = \left[\sum_{k=1}^n |q_k - r_k|^p \right]^{1/p}$ for $p \geq 1$, where $Q \equiv (q_1, \dots, q_n)$ and $R \equiv (r_1, \dots, r_n)$ define the locations of the new and existing facilities. When $p = 1$, we obtain the results for the rectilinear distance case and when $p = 2$ we solve the Euclidean distance problem. No interaction between new facilities is considered here.

1.3.2 Location-Allocation Problems

When both the location of new facilities and the allocations of products are decision variables, the degree of complexity increases appreciably. It comes as no surprise then that most of the contributions

to literature on this subject deal with heuristic procedures. We shall classify some of the available techniques according to whether Rectilinear or Euclidean distance measure is employed.

1. Rectilinear Distance Location Allocation Problems. In most practical situations, the allocation of flow between facilities depends on the relative location of these facilities and conversely, the location of new facilities depends on the extent of interaction between new and existing facilities. Cooper (9) uses this concept in order to heuristically arrive at a solution. He iterates between the pure allocation and the pure location problems, in each case using the result of the previous iteration until the solution converges. Such a procedure can easily converge to a non-optimal point. This is demonstrated in reference (37).

Cooper (9) provides another heuristic procedure to an uncapacitated problem of this type. For the purpose of solution, the capacities of the sources are fixed at an arbitrarily large value. The algorithm then adjusts the allocations at each step of the iterative procedure by comparing the requirements at an existing facility with the surplus in the capacity of the new facility. It is argued that this problem is more realistic because in addition to locating the new facilities, the solution also yields the optimum capacity it should have. This provides a helpful fact in the design of the new facilities.

Cooper (9) also provides an exact algorithm which amounts to determining all basic feasible solutions corresponding to every extreme point of the pure location problem and then solving the pure allocation problem for each of these solutions. The optimum solution is obtained

by selecting the best from among all these solutions.

Morris (31) provides an interesting exact approach to solve the rectilinear location-allocation problem. He identifies the extreme points of the location problem and formulates the problem using zero-one variables z_{ik} to indicate whether the new facility i is assigned to the k^{th} extreme point and zero-one variables z_{ij} to indicate whether new facility i interacts with existing facility j . Morris assumes that $\sum_{i=1}^n z_{ij} = 1$ for all j and his approach is similar to that of Effroymsen and Ray (12) which we introduced earlier. The claim that the solutions are always integer valued is backed by the computational experience of Revelle and Swain (36). However, if there are m existing facilities then there are $m^3 + m^2 - m$ variables and $m^3 + 1$ constraints. This limits m to 15 when using the LP1108 code which can accommodate 4044 rows and 99000 columns. The advantage is that the model is computationally insensitive to the number of new facilities n which occurs just once in one of the constraints. Hence the problem of optimizing over n can be solved with great ease using the basis inverse obtained from the previous optimum solution. Kuenne and Soland (22) worked similarly using a branch and bound algorithm.

A more efficient solution procedure to this problem was provided by Sherali (37). He iterated between the pure allocation and location problems as Cooper did in his heuristic procedure, but identified the resulting solution rightly as a local optimal. He then introduced cutting planes to work towards the global optimum. We will work similarly but will show how deeper cuts may be generated in an even more efficient manner. Also, unlike as in (37), we will develop a procedure which is

guaranteed to determine a point in the solution space which is feasible to the cuts generated.

2. Euclidean Distance Location-Allocation Problems. Cooper (10) has presented some heuristic procedures for this class of problems also. He tackles the problem of equal requirements and unlimited capacity first. The allocation part is trivially solved and for the location of a single new facility, he picks the closest pair of existing facilities each time and eliminates the one with the greatest sum of the distances from the other existing facilities. He also provides heuristic procedures for the unequal requirements-unlimited capacity and the unequal requirements - limited capacity problems. In the last case, he claims that convergence to a local optimum usually with 2 to 3% of the global optimum is obtained, the maximum expected error being about 10%.

1.4 Summary of Proposed Approach

The literature search has revealed that the problem at hand has not as yet been tackled adequately. In fact, there exists not even a satisfactory heuristic procedure to solve it.

We will first recast our problem into the framework of a Bi-linear Programming problem and investigate its properties such as non-convexity and extreme point optimality. We will then develop an algorithm for the interacting multi-facility location problem which will be a subroutine in our solution procedure. We will then develop a procedure to find a local optimum.

Thereafter, we will develop an algorithm to generate efficiently cutting planes which are deeper than those obtained by other published

algorithms. We will also develop an algorithm which will find a solution feasible to the set of cuts we have generated and will proceed to introduce additional cuts until the feasible region is exhausted.

The entire procedure will be coded and run on a CDC Cyber 74 machine. We will test the computational efficiency of both the pure location problem as well as the location-allocation problem against existing procedures. We will also indicate the limiting size problem which this procedure can handle.

CHAPTER II

BASIC PROPERTIES OF THE GENERAL RECTILINEAR-DISTANCE LOCATION-ALLOCATION PROBLEM (GRLAP)

2.1 Introduction

In this chapter, we will investigate some fundamental characteristics of our problem which will be useful in this study. We will first cast our problem into the general framework of a Bilinear Programming Problem (BLP) and use some definitions and theorems available for the latter class of problems to study the nature of our problem. We will then introduce some theorems to characterize optimality for our problem. Finally, to aid us in applying these theorems, we will partition GRLAP into its allocation and location aspects.

2.2 Transformation of GRLAP into BLP

The Bilinear Programming Problem has the form

PROBLEM BLP:

$$\underline{\min.} \quad \phi(z,u) = c^t z + d^t u + z^t D u$$

$$\underline{\text{subj. to}} \quad z \in Z$$

$$u \in U$$

where, Z and U are non-empty bounded polyhedral sets and c , d and D are appropriate coefficient matrices.

The Bilinear Programming Problem gets its name from the fact that if the values of either the set of variables z (or u) are specified, it reduces to a linear programming problem in the other set of variables.

We will now use a simple transformation which we will validate at the end of this chapter. Let,

$$x_i - x_j = x_{ij}^+ - x_{ij}^- \quad \text{for } i=1, \dots, n \text{ and } j=1, \dots, n+m$$

$$x_{ij}^+, x_{ij}^- \geq 0, x_{ij}^+ \cdot x_{ij}^- = 0$$

and similarly,

$$y_i - y_j = y_{ij}^+ - y_{ij}^- \quad \text{for } i=1, \dots, n \text{ and } j=1, \dots, n+m$$

$$y_{ij}^+, y_{ij}^- \geq 0, y_{ij}^+ \cdot y_{ij}^- = 0$$

Then,

$$|x_i - x_j| = x_{ij}^+ + x_{ij}^-$$

and

$$|y_i - y_j| = y_{ij}^+ + y_{ij}^-$$

We may now write problem GRLAP as

PROB. GRLAP 1:

$$\underline{\min.} \quad \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^{n+m} c_{ik} u_{ijk} + \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^{n+m} t_k (x_{ij}^+ + x_{ij}^- + y_{ij}^+ + y_{ij}^-) u_{ijk}$$

subj. to Constraint set $C_1 \equiv$

$$\sum_{j=1}^{n+m} u_{ijk} \leq a_{ik} \quad i=1, \dots, n; k=1, \dots, K$$

$$\sum_{i=1}^n u_{ijk} = b_{jk} \quad j=1, \dots, n+m; k=1, \dots, K$$

$$u_{ijk} \geq 0 \quad i=1, \dots, n; j=1, \dots, n+m, k=1, \dots, K$$

and Constraint set $C_2 \equiv$

$$x_i - x_j - x_{ij}^+ + x_{ij}^- = 0 \quad i, j = 1, \dots, n$$

$$y_i - y_j - y_{ij}^+ + y_{ij}^- = 0 \quad i, j = 1, \dots, n$$

$$x_i - x_{ij}^+ + x_{ij}^- = x_j \equiv d_j \quad i = 1, \dots, n \text{ and } j = n+1, \dots, n+m$$

$$y_i - y_{ij}^+ + y_{ij}^- = y_j \equiv e_j \quad i = 1, \dots, n \text{ and } j = n+1, \dots, n+m$$

$$x_{ij}^+ \cdot x_{ij}^- = 0, \quad y_{ij}^+ \cdot y_{ij}^- = 0 \quad i = 1, \dots, n; j = 1, \dots, n+m$$

$$x_i, y_i, x_{ij}^+, x_{ij}^-, y_{ij}^+, y_{ij}^- \geq 0 \quad i = 1, \dots, n; j = 1, \dots, n+m$$

In the next chapter, we will demonstrate that the problem solution is not affected by ignoring the constraints $x_{ij}^+ \cdot x_{ij}^- = 0$ and $y_{ij}^+ \cdot y_{ij}^- = 0$ since those are automatically satisfied. We will hence disregard these constraints here.

We now define sets Z and U as:

$$\begin{aligned} Z = \{z \equiv & (x_{11}^+, \dots, x_{1,n+m}^+, x_{21}^+, \dots, x_{2,n+m}^+, \dots, x_{n1}^+, \dots, x_{n,n+m}^+, \\ & x_{11}^-, \dots, x_{n,n+m}^-, y_{11}^+, \dots, y_{n,n+m}^+, y_{11}^-, \dots, y_{n,n+m}^-, \\ & x_1, \dots, x_n, y_1, \dots, y_n)^t: \text{CONSTRAINT SET} \\ & C_2 \text{ IS SATISFIED}\} \end{aligned}$$

and

$$\begin{aligned} U = \{u \equiv & (u_{1,1,1}, \dots, u_{1,n+m,1}, u_{2,1,1}, \dots, u_{2,n+m,1}, \dots, \\ & u_{n,1,1}, \dots, u_{n,n+m,1}, u_{1,1,2}, \dots, u_{n,n+m,2}, \dots, \\ & u_{1,1,k}, \dots, u_{n,n+m,k})^t: \text{CONSTRAINT SET} \\ & C_1 \text{ IS SATISFIED}\} , \end{aligned}$$

We also define matrices D and d as:

$$d^t = (c_{11} \cdot 1_{n+m}^t, c_{21} \cdot 1_{n+m}^t, \dots, c_{n1} \cdot 1_{n+m}^t, c_{12} \cdot 1_{n+m}^t, \dots, c_{n2} \cdot 1_{n+m}^t, \dots, c_{nK} \cdot 1_{n+m}^t)$$

where 1_{n+m}^t is a row vector of $(n+m)$ elements, all equal to 1.

and

$$D = \begin{bmatrix} t_1^t 1_{n(m+n) \times n(m+n)} & \cdot & \cdot & \cdot & t_K^t 1_{n(m+n) \times n(m+n)} \\ t_1^t 1_{n(m+n) \times n(n+m)} & \cdot & \cdot & \cdot & t_K^t 1_{n(m+n) \times n(n+m)} \\ t_1^t 1_{n(m+n) \times n(m+n)} & \cdot & \cdot & \cdot & t_K^t 1_{n(m+n) \times n(m+n)} \\ t_1^t 1_{n(m+n) \times n(m+n)} & \cdot & \cdot & \cdot & t_K^t 1_{n(m+n) \times n(m+n)} \\ 0_{2n \times nK(n+m)} \end{bmatrix}$$

The size of D is $[2n(2n+2m+1) \times nK(n+m)]$.

Problem GRLAP1 may then be rewritten compactly as:

PROB. P_1

$$\underline{\min.} \quad \Phi(z, u) = d^t u + z^t D u$$

$$\underline{\text{subj. to}} \quad z \in Z$$

$$u \in U$$

(Note that sets Z and U are polyhedral sets since all the constraints defining them are linear).

We observe that problem P_1 has the exact form of problem BLP with $c = 0$.

Problem P_1 may also be expressed as the quadratic form:

PROB. P_2

$$\min. \quad \Phi(p) = q^t p + \frac{1}{2} p^t C p$$

$$p \in Z \times U$$

where,

$$q^t = (0, d^t), \quad p^t = (z^t, u^t) \quad \text{and} \quad C = \begin{bmatrix} 0 & D \\ D^t & 0 \end{bmatrix}$$

2.3 Fundamental Definitions and Theorems

In this section, we will present some definitions and theorems (mainly Martos' Theorems) to assist us in our study.

Definition 2.3.1. Let $f: E_n \rightarrow E_1 \cup \{\infty\}$ and let $\text{domain } f = \{x: f(x) < \infty\}$. f is called a convex (concave) function if $f(\lambda x_1 + (1-\lambda)x_2) \leq (>) \lambda f(x_1) + (1-\lambda)f(x_2)$ for each $x_1, x_2 \in \text{domain of } f$ and for each $0 \leq \lambda \leq 1$.

Definition 2.3.2. A function f defined over a convex set S is quasi concave on S if for all $x_1, x_2 \in S$,

$$f[\lambda x_1 + (1-\lambda)x_2] \geq \min. [f(x_1), f(x_2)] \quad \text{for } 0 \leq \lambda \leq 1.$$

The negative of a quasi-concave function is quasi-convex.

Definition 2.3.3. A quasi-convex function f defined over a convex set S is explicitly quasi-convex on S if for all $x_1, x_2 \in S$ with $f(x_1) \neq f(x_2)$,

$$f[\lambda x_1 + (1-\lambda)x_2] < \max. [f(x_1), f(x_2)]$$

$$\text{for } 0 < \lambda < 1.$$

Definition 2.3.4. An $(n \times n)$ matrix D is positive subdefinite if $x^t D x < 0$ implies that either $Dx \geq 0$ or $Dx \leq 0$ for any vector $x \in E_n$. A quadratic form $\phi(x) = x^t D x$ is said to be positive subdefinite if D is positive subdefinite.

Theorem 2.3.5. The quadratic form $\phi(x) = x^t D x$ is quasi-convex on the non-negative orthant, E_n^+ , if and only if it is positive subdefinite.

Proof: See (29).

Theorem 2.3.5

A continuous function f defined over a polytope L attains its minimum at an extreme point of L and of all its convex polyhedral subsets if and only if it is quasi-concave on L .

Proof: See (30).

Theorem 2.3.6.

A continuous function f defined over a polytope L is such that each local minimum is also a global minimum on L and on all its convex polyhedral subsets if and only if f is explicitly quasi-convex on L .

Proof: See (30).

2.4 Investigation of Some Basic Properties of Problems P_1 and P_2

Theorems 2.3.5 and 2.3.6 indicate that quasi-concavity and quasi-convexity are desirable properties in developing an efficient solution procedure. However, our problem possesses neither of these properties and we will demonstrate this through the following counter examples.

Lemma 2.4.1 The objective function $\phi(z, u)$ of P_1 (or equivalently $\phi(p)$ of problem P_2) is not quasi-convex on the non-negative orthant E_n^+ .

Proof: Let $K = 1$, $t_1 = 1$, $c_{i1} = c$ for all $i = 1, \dots, n$.

Let $n = m = 2$. Hence,

$$\sum_{i=1}^n \sum_{j=1}^{n+m} \sum_{k=1}^K c_{ik} u_{ijk} = c \sum_{i=1}^n \sum_{j=1}^{n+m} u_{ij1} = c \sum_{j=1}^{n+m} b_{j1}$$

which is a constant since demands b_{j1} are specified.

Hence, the objective functions of P_1 and P_2 become

$$\min. \phi(z, u) = z^t D u \quad \text{and}$$

$$\min. \phi(p) = \frac{1}{2} p^t C p \quad \text{respectively.}$$

Consider $p^t = (0, 0, 0, 0, 2, 0, 2, 0, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, -1, 1, 0, 1, 0, 1, -4, 0, 0, 0, 0, 0)_{1 \times 44}$.

In P_1 , D has dimensions 36×8 and C is therefore 44×44 in P_2 . Also

$$\frac{1}{2} p^t C p = (-1) < 0 \quad \text{and} \quad C p = (0, 1, -4, 0, 0, 0, 0, 0, 0, 1, -4, 0, 0, 0, 0, 0, 0, 1, -4, 0, 0, 0, 0, 0, 0, 1, -4, 0, 0, 0, 0, 0, 1, -4, 0, 0, 0, 0, 0, 0, 1, -4, 0, 0, 0, 0, 0, 0, 0, 3, 1, 2, 3, 0, 2, 1)^t_{(44 \times 1)}$$

We hence observe that $\frac{1}{2} p^t C p < 0$ but $C p$ is neither ≥ 0 nor ≤ 0 . Hence, C is not positive subdefinite and hence, $\phi(z, u)$ is not quasi-convex over the non-negative orthant E_n^+ .

Lemma 2.4.2 $\phi(z, u)$ is not quasi-concave on the non-negative orthant E_n^+ .

Proof: Let $K = 1$, $t_1 = 1$, $c_{i1} = c$ (for all i), $n = m = 2$.

$$\text{Let } p^t = (0, 0, 0, 0, 2, 0, 2, 0, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, -1, 1, 0, 1, 0, -1, 4, 0, 0, 0, 0, 0)_{1 \times 44}.$$

$$\text{Hence, } C p = (0, -1, 4, 0, 0, 0, 0, 0, 0, -1, 4, 0, 0, 0, 0, 0, 0, -1, 4, 0, 0, 0, 0, 0, 0, -1, 4, 0, 0, 0, 0, 0, 0, -1, 4, 0, 0, 0, 0, 0, 0, 3, 1, 2, 3, 0, 2, 1)^t.$$

Hence, $\frac{1}{2} p^t C p = 1 > 0$, but $C p$ is neither ≥ 0 nor ≤ 0

$\therefore C$ is not negative sub-definite. $\therefore \phi(z, u)$ is not quasi-concave over

the non-negative orthant E_n^+ .

These counter examples show that we cannot invoke Theorem 2.3.6 to our advantage. Also, Theorem 2.3.5 is violated. However, we will now proceed to demonstrate that our problem has the special property that the optimum occurs at an extreme point of its constraint set. We will take advantage of this in solving our problem. We will also attempt to account for this apparent violation of Theorem 2.3.5.

Theorem 2.4.3. The solution (\bar{z}, \bar{u}) is an extreme point of the set $S = \{(z, u) : (z, u) \in (Z \times U)\}$ if and only if \bar{z} is an extreme point of Z and \bar{u} is an extreme point of U .

Proof: See (41).

Corollary 2.4.4. Each adjacent extreme point of $(\bar{z}, \bar{u}) \in S$ is either of the form (z^i, \bar{u}) where $z^i \in A(\bar{z})$, or of the form (\bar{z}, u^i) where $u^i \in A(\bar{u})$, where $A(\bar{z})$ and $A(\bar{u})$ are the sets of adjacent extreme points of \bar{z} in Z and \bar{u} in U , respectively.

Proof: See (41).

Definition 2.4.5. The function $\phi(z, u)$ defined over the polyhedral sets Z and U has a local star minimum at the point $(\bar{z}, \bar{u}) \in Z \times U$ if $\phi(\bar{z}, \bar{u}) \leq \phi(z, u)$ for all $(z, u) \in A(\bar{z}, \bar{u})$ where $A(\bar{z}, \bar{u})$ represents the set of adjacent extreme points of (\bar{z}, \bar{u}) .

Theorem 2.4.6. (\bar{z}, \bar{u}) is a local star minimum of problem P_1 if and only if for a fixed u , \bar{z} is a solution to:

$$P_{11} \text{ :- } \begin{array}{ll} \min. & \phi(z, \bar{u}) = z^t D \bar{u} \\ \text{subj. to} & z \in Z \end{array}$$

and for a fixed \bar{z} , \bar{u} is a solution to

$$P_{12} := \begin{array}{ll} \min. & \phi(\bar{z}, u) = d^t u + \bar{z}^t D u \\ \text{subj. to} & u \in U. \end{array}$$

Proof: See (41).

2.4.7 Algorithm for a Local Star Minimum

Based on theorem 2.4.6, the following procedure to arrive at a local star minimum is evident:

Step 1: Start with an extreme point of the set Z , say $\bar{z} = z^0$. Solve problem P_{12} to obtain a solution u^0 .

Step 2: Solve P_{11} using $\bar{u} = u^0$ and obtain a solution z^1 . If $z^1 = z^0$, stop. Else, put $\bar{z} = z^1$ and go to step 1.

Finite convergence is proved in reference (37).

Definition 2.4.8. The function $\phi(z, u)$ defined over $Z \times U$ has a global minimum at the point (\bar{z}, \bar{u}) if $\phi(\bar{z}, \bar{u}) \leq \phi(z, u)$ for each $(z, u) \in (Z \times U)$.

Definition 2.4.9. The extreme point (\bar{z}, \bar{u}) of $Z \times U$ is called a pseudo-global minimum if

$$\min_{u \in U} \phi(z^i, u) \geq \min_{u \in U} \phi(\bar{z}, u) = \phi(\bar{z}, \bar{u}) \text{ for each } z^i \in A(\bar{z}), \text{ where } A(\bar{z}) \text{ is the set of adjacent extreme points of } \bar{z} \text{ in } Z.$$

Lemma 2.4.10. Problem P_1 has an optimum solution (\bar{z}, \bar{u}) which is such that \bar{z} is an extreme point of Z and \bar{u} is an extreme point of U .

Proof: For an arbitrary $u^1 \in U$, consider

$$\begin{array}{ll} \min. & \phi(z, u^1) = (d^t u^1 + z^t D u^1) \\ \text{subj. to} & z \in Z. \end{array}$$

This is a linear programming problem and it therefore has an extreme point solution \bar{z} with

$$\phi(\bar{z}, u') \leq \phi(z, u') \quad \forall z \in Z$$

Consider now, the problem

$$\begin{array}{ll} \min. & \phi(\bar{z}, u) = d^t u + \bar{z}^t u \\ \text{subj. to} & u \in U. \end{array}$$

Again, this is an l.p. problem and hence has an optimal extreme point solution \bar{u} such that $\phi(\bar{z}, \bar{u}) \leq \phi(\bar{z}, u) \quad \forall u \in U$. Hence, $\phi(\bar{z}, \bar{u}) \leq \phi(\bar{z}, u') \leq \phi(z, u')$. Since, we can repeat this for each such $u' \in \mathcal{U}$ (finite in number), there exists a \bar{z} , an extreme point of Z and \bar{u} an extreme point of \mathcal{U} such that $\phi(\bar{z}, \bar{u}) \leq \phi(\bar{z}, u) \leq \phi(z, u) \quad \forall (z, u) \in (Z \times U)$.

It may appear that this theorem violates Martos' theorem 2.3.5 in view of the non-quasi-concavity of $\phi(z, u)$ over the non-negative orthant E_n^+ . However, this is not so because Theorem 2.3.5 stipulates that the function must attain its optimum at an extreme point of all convex polyhedral subsets of its constraint set L . For problem P_1 , however, it is evident that if we have constraints involving both z and u the proof of theorem 2.4.10 does not hold. Hence due to non-quasi-concavity, although $\phi(z, u)$ obtains its minimum at an extreme point of $Z \times U$, it does not do so over all convex polyhedral subsets of this constraint set.

It may be noted that Problem P_1 may have a local star minimum different from the global minimum. The illustrative examples of Chapter V illustrate this fact.

2.5. The Allocation and Location Aspects of Problem GRLAP1

Throughout this chapter, we have referred to the solutions of problem

P_1 with either z or u fixed. The solution procedure for such problems is discussed below:

2.5.1 Problem with Fixed z

In problem GRLAPI (from which problem P_1 was derived), if we are specified the values of (x_i, y_j) $i = 1, \dots, n+m$, we obtain:

Problem TRANS:

$$\begin{aligned} \min. \quad & \sum_{k=1}^k \left[\sum_{i=1}^n \sum_{j=1}^{n+m} \{c_{ijk} + t_k(x_{ij}^+ + x_{ij}^- + y_{ij}^+ + y_{ij}^-)\} u_{ijk} \right] \\ \text{subj. to} \quad & \left. \begin{aligned} \sum_{j=1}^{n+m} u_{ijk} &\leq a_{ik} \quad \text{for } i = 1, \dots, n \\ \sum_{i=1}^n u_{ijk} &= b_{jk} \quad \text{for } j = 1, \dots, n+m \\ u_{jk} &\geq 0 \quad \forall i, j \end{aligned} \right\} \text{for } k = 1, \dots, K. \end{aligned}$$

It is easy to see that these are K transportation problems for which some very efficient solution techniques are available (39).

2.5.2 Problem with Fixed u

If the allocations are specified, we obtain

$$\text{PROB. MFLOC:- } \min. \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^{n+m} t_k u_{ijk} (x_{ij}^+ + x_{ij}^- + y_{ij}^+ + y_{ij}^-)$$

$$\text{subj. to } x_i - x_j = x_{ij}^+ - x_{ij}^- \quad i = 1, \dots, n, \quad j = 1, \dots, n+m$$

$$x_i, x_j, x_{ij}^+, x_{ij}^- \geq 0, \quad x_{ij}^+ \cdot x_{ij}^- = 0$$

and

$$y_i - y_j = y_{ij}^+ - y_{ij}^-$$

$$y_i, y_j, y_{ij}^+, y_{ij}^- \geq 0, \quad y_{ij}^+ \cdot y_{ij}^- = 0 \quad i = 1, \dots, n$$

$$j = 1, \dots, n+m.$$

It is easily seen that this problem is decomposable into two problems, one involving the x coordinates alone and the other in y . These two problems are the multifacility location problems for which we will develop an algorithm in the next chapter.

Hill and Ravindran (16) pointed out that for a problem of the type

$$\begin{aligned} \underline{\max.} \quad z &= \sum_j c_j |x_j| \\ Ax &= b \\ x &\text{ unrestricted} \end{aligned}$$

an l.p. simplex method based solution using the transformation $x_j = x_j^+ - x_j^-$, $x_j^+ \geq 0$, $x_j^- \geq 0$, ($x_j^+ \cdot x_j^- = 0$ being automatically satisfied) may give an unbounded solution even if the original problem has a finite optimum. A necessary and sufficient condition for a finite optimum was shown to be $Ay = 0 \Rightarrow c|y| \leq 0$. (The components of $|y|$ are the absolute values of those of y .)

Since the transformed problem MFL0C clearly has a lower bound of zero, a finite optimum is guaranteed.

CHAPTER III

THE INTERACTING MULTIFACILITY LOCATION PROBLEM

3.1 Introduction

In this chapter, we will develop an algorithm to solve problem MFLOC introduced in Section 2.5.2. An x coordinate solution will be provided and needless to say, a similar approach holds for the y coordinates. We will first reduce the dimensionality of this problem by suitable redefinitions and then investigate its characteristics. An algorithm will be developed to solve the problem in two steps (i) Moving one facility at a time to give a strict improvement in the objective function value and (ii) considering joint movements of facilities to yield a strict improvement of the objective function. Two illustrative examples will be provided to clarify the procedure.

3.2 Problem Reformulation and Description

Consider Problem MFLOC_x - Letting $w_{ij}^i = \sum_{k=1}^K t_k u_{ijk}$, we have,

$$\underline{\text{min.}} \quad f = \sum_{i=1}^n \sum_{j=1}^{n+m} w_{ij}^i (x_{ij}^+ + x_{ij}^-)$$

$$\underline{\text{subj. to}} \quad x_i - x_j - x_{ij}^+ + x_{ij}^- = 0$$

$$x_{ij}^+ \cdot x_{ij}^- = 0$$

$$x_i, x_j, x_{ij}^+, x_{ij}^- \geq 0$$

for $i = 1, \dots, n$
and $j = 1, \dots, n+m$.

3.2.1 Lemma. For any feasible solution, problem MFLOC_x will have

$$x_{ij}^+ = x_{ji}^- \text{ and } x_{ij}^- = x_{ji}^+ \text{ for all } i, j = 1, \dots, n.$$

Proof. Assume without loss of generality that $x_i \leq x_j$.

By definition,

$$\begin{aligned} x_i - x_j + x_{ij}^+ - x_{ij}^- &\leq 0 \\ x_{ij}^+ &\geq 0, \quad x_{ij}^- \geq 0, \quad x_{ij}^+ \cdot x_{ij}^- = 0. \end{aligned}$$

This implies that

$$x_{ij}^- = -(x_i - x_j) \text{ and } x_{ij}^+ = 0 \quad (1)$$

$$\text{Also, } x_j - x_i = x_{ji}^+ - x_{ji}^- \geq 0,$$

$$x_{ji}^+ \geq 0, \quad x_{ji}^- \geq 0, \quad x_{ji}^+ \cdot x_{ji}^- = 0.$$

This implies that

$$x_{ji}^+ = -(x_i - x_j) \text{ and } x_{ji}^- = 0 \quad (2)$$

Comparing (1) and (2), we have,

$$x_{ij}^+ = x_{ji}^- \text{ and } x_{ij}^- = x_{ji}^+.$$

Using the above lemma, the objective function

$$f = \sum_{i=1}^n \sum_{j=1}^n w_{ij}^1 (x_{ij}^+ + x_{ij}^-) + \sum_{i=1}^n \sum_{j=n+1}^{n+m} w_{ij}^1 (x_{ij}^+ + x_{ij}^-)$$

of problem MFLOC_x can be simplified. Let

$$z_1 = \sum_{i=1}^n \sum_{j=1}^n w_{ij}^1 (x_{ij}^+ + x_{ij}^-).$$

Then noting that $w_{ii}^1 = 0$, we have,

$$\begin{aligned}
z_1 &= \sum_{i=1}^n \sum_{i < j \leq n} w_{ij}^! (x_{ij}^+ + x_{ij}^-) + \sum_{i=2}^n \sum_{1 \leq j < i \leq n} w_{ij}^! (x_{ij}^+ + x_{ij}^-) \\
&= \sum_{i=1}^n \sum_{i < j \leq n} w_{ij}^! (x_{ij}^+ + x_{ij}^-) + \sum_{j=2}^n \sum_{1 \leq i < j \leq n} w_{ji}^! (x_{ji}^+ + x_{ji}^-)
\end{aligned}$$

Noting the values taken by i and j in the second term, we can readjust the order of summation to get

$$z_1 = \sum_{i=1}^n \sum_{i < j \leq n} w_{ij}^! (x_{ij}^+ + x_{ij}^-) + \sum_{i=1}^n \sum_{i < j \leq n} w_{ji}^! (x_{ji}^+ + x_{ji}^-)$$

We now use Lemma 3.2.1 on the second term to obtain,

$$\begin{aligned}
z_1 &= \sum_{i=1}^n \sum_{i < j \leq n} w_{ij}^! (x_{ij}^+ + x_{ij}^-) + \sum_{i=1}^n \sum_{i < j \leq n} w_{ji}^! (x_{ij}^- + x_{ij}^+) \\
&= \sum_{i=1}^n \sum_{i < j \leq n} (w_{ij}^! + w_{ji}^!) (x_{ij}^+ + x_{ij}^-).
\end{aligned}$$

Letting $(w_{ij}^! + w_{ji}^!) = w_{ij}$, we get

$$z_1 = \sum_{i=1}^n \sum_{i < j \leq n} w_{ij} (x_{ij}^+ + x_{ij}^-).$$

Also, letting $w_{ij}^! = w_{ij}$ for $i=1, \dots, n$ and $j = n+1, \dots, n+m$, Problem MFLOC_x becomes,

$$\begin{aligned}
\underline{\text{PROBLEM P}}_x^! &= \underline{\text{min.}} \quad f = \sum_{i=1}^n \sum_{i < j \leq n+m} w_{ij} (x_{ij}^+ + x_{ij}^-) \\
&\quad \underline{\text{subj. to}} \quad x_i - x_j - x_{ij}^+ + x_{ij}^- = 0 \quad \{i=1, \dots, n; j=i, i+1, \dots, n+m\} \\
&\quad x_i, x_j, x_{ij}^+, x_{ij}^- \geq 0 \quad \forall i, j \\
&\quad x_{ij}^+ \cdot x_{ij}^- = 0 \quad \{i=1, \dots, n; j=i, i+1, \dots, n+m\}
\end{aligned}$$

We may now observe that without the constraints $x_{ij}^+ \cdot x_{ij}^- = 0$, the problem P_x^I is a linear programming problem. Also, for each i and j , x_{ij}^+ and x_{ij}^- appear in only one constraint and their "columns" being linearly dependent, they cannot simultaneously be basic and hence, $x_{ij}^+ \cdot x_{ij}^-$ will equal zero for all basic feasible solutions of problem P_x^I without these constraints.

3.2.2 Transformation for the Case where more Than One Existing Facility has the Same x-Coordinate

Let us assume that $d_a = d_b$ for $a, b \in \{n+1, \dots, n+m\}$. In the constraint set of Problem P_x^I , for $i=1, \dots, n$, we have,

$$x_i - x_{ia}^+ + x_{ia}^- = x_a$$

and
$$x_i - x_{ib}^+ + x_{ib}^- = x_b = x_a.$$

$$\therefore (x_{ia}^+ - x_{ia}^-) = (x_{ib}^+ - x_{ib}^-) = (x_i - x_a)$$

Hence,
$$x_{ia}^+ = x_{ib}^+ = x_i - x_a \quad \text{if } x_i > x_a$$

$$= 0 \quad \text{otherwise}$$

and
$$x_{ia}^- = x_{ib}^- = x_a - x_i \quad \text{if } x_i < x_a$$

$$= 0 \quad \text{otherwise.}$$

In the objective function of Problem P_x^I , terms involving \underline{a} and \underline{b} may be transformed as

$$\sum_{i=1}^n [w_{ia}(x_{ia}^+ + x_{ia}^-) + w_{ib}(x_{ib}^+ + x_{ib}^-)]$$

$$= \sum_{i=1}^n (w_{ia} + w_{ib})(x_{ia}^+ + x_{ia}^-) .$$

The above expression indicates that facilities \underline{a} and \underline{b} can be replaced by a single (existing) facility \bar{a} such that $w_{i\bar{a}} = (w_{ia} + w_{ib})$. The corresponding constraint will then be,

$$-x_{i\bar{a}}^+ + x_{i\bar{a}}^- + x_i = x_{\bar{a}} \quad \text{for } i = 1, \dots, n$$

where $x_{\bar{a}} = x_a = x_b$,

and the term in the objective function will be

$$\sum_{i=1}^n w_{i\bar{a}} (x_{i\bar{a}}^+ + x_{i\bar{a}}^-).$$

The effect of the above transformation is to essentially make the two existing facilities a and b into a single facility \bar{a} . Henceforth, we will assume that such transformations are made prior to applying the algorithm we will develop for the location problem. Besides reducing the number of variables and constraints, this will enable us to renumber the existing facilities so that they are ordered relative to the coordinates we are currently working with. Hence for problem P'_x , $x_{n+1} < x_{n+2} < \dots < x_{n+m}$ where m is now the effective number of existing facilities we have for the "x coordinate problem."

3.2.3 Final Transformation

In problem P'_x (without the constraints $x_{ij}^+ \cdot x_{ij}^- = 0$) we now multiply each constraint by its associated w_{ij} and subtract it from the objective function. (We can do this because we have equality constraints). The modified objective function is

$$\min. \quad f = \sum_{i=1}^n \sum_{i < j \leq n+m} (2w_{ij} \cdot x_{ij}^+) - \sum_{i=1}^n w_i x_i + \sum_{j=n+1}^{n+m} x_j \left(\sum_{i=1}^n w_{ij} \right)$$

where,

$$w_i = \left(\sum_{j=i+1}^{n+m} w_{ij} - \sum_{t=1}^{i-1} w_{ti} \right)$$

The last term in f is a constant since it is simply the sum of the products of the x coordinate of each existing facility with its total demand.

We may now state our problem as:

PROBLEM P_x

$$\begin{aligned} \min. \quad & \sum_{i=1}^n \sum_{i < j \leq n+m} 2w_{ij} x_{ij}^+ - \sum_{i=1}^n w_i x_i \\ \text{subj. to} \quad & -x_{ij}^+ + x_{ij}^- + x_i - x_j = 0 \quad (i=1, \dots, n; j=i+1, \dots, n) \\ & -x_{ij}^+ + x_{ij}^- + x_i = x_j = d_j \quad (i=1, \dots, n; j=n+1, \dots, n+m) \\ & x_{ij}^+, x_{ij}^-, x_i, x_j \geq 0 \quad \forall i, j \end{aligned}$$

3.2.4 Lemma Each basic feasible solution to P_x will have $x_i = d_j$

for each $i = 1, \dots, n$ and $j \in \{n+1, \dots, n+m\}$.

Proof: See (34) and (8).

3.2.5 Lemma. A simplex pivot operation in the tableau representing problem P_x , would not involve the movement of a new facility at a given location to a non-adjacent location. (Here, by an adjacent location, we mean an existing facility either to the left or to the right of that at which the new facility is currently placed).

Proof: See (35).

3.2.6 Lemma. Each basic feasible solution to P_x will have $m \cdot n + \frac{n(n-1)}{2}$ basic variables.

Proof: Let us first determine the number of constraints in Problem P_x (aside from the non-negativity constraints). (i) There is one constraint for every pair $i, j \in \{1, \dots, n\}$ ($i \neq j$). Hence there are $\binom{n}{2} = \frac{n(n-1)}{2}$ such constraints. (ii) Also, for each combination of $i=1, \dots, n$ and $j = n+1, \dots, n+m$, there is one constraint. This gives us another $m \cdot n$ constraints. There are hence $m \cdot n + \frac{n(n-1)}{2}$ constraints. If this constraint set is represented as $\underline{A} \underline{x} = \underline{b}$, then the coefficient matrix \underline{A} has rank $\leq m \cdot n + \frac{n(n-1)}{2}$. We will now demonstrate that \underline{A} has full rank. For this, we partition out a square matrix of that size from \underline{A} and show that it has a non-zero determinant. Specifically, let us select the coefficient columns for all the x_{ij}^- variables ($i=1, \dots, n$ and $j=i+1, \dots, n+m$). Each x_{ij}^- appears in only one constraint and hence we have selected an identity matrix of size $[m \cdot n + \frac{n(n-1)}{2}] \times [m \cdot n + \frac{n(n-1)}{2}]$. Since this has a determinant value equal to one, \underline{A} has full rank.

3.2.7 Classification of Basic Variables into Sets

We have just seen that the total number of basic variables is $m \cdot n + \frac{n(n-1)}{2}$. We will now define mutually exclusive and collectively exhaustive sets such that given a basic feasible solution, we can classify the basic variables according to these sets. This will be helpful to us in later sections.

For the sake of convenience, we will refer to the pair (x_{ij}^+, x_{ij}^-) as x_{ij} for all i, j . Noting that both x_{ij}^+ and x_{ij}^- cannot be basic (as established above), we define the following sets: $S_1 = \{x_1, \dots, x_n\}$. By

a suitable choice of the origin, we can ensure that $d_1, \dots, d_m > 0$ and by Lemma 3.2.4, $x_1, \dots, x_n > 0$ and hence, x_1, \dots, x_n are always basic.

$$S_2 = \{x_{ij} : x_{ij}^+ \text{ is basic, } x_{ij}^- \text{ is nonbasic, } i \in \{1, \dots, n\}, j \in \{i+1, \dots, n\}\}$$

$$S_3 = \{x_{ij} : x_{ij}^- \text{ is basic, } x_{ij}^+ \text{ is nonbasic, } i \in \{1, \dots, n\}, j \in \{i+1, \dots, n\}\}$$

$$S_4 = \{x_{ij} : x_{ij}^+ \text{ and } x_{ij}^- \text{ are nonbasic, } i \in \{1, \dots, n\}, j \in \{i+1, \dots, n\}\}$$

$$S_5 = \{x_{ij} : x_{ij}^+ \text{ is basic, } x_{ij}^- \text{ is nonbasic, } i \in \{1, \dots, n\}, j \in \{n+1, \dots, n+m\}\}$$

$$S_6 = \{x_{ij} : x_{ij}^- \text{ is basic, } x_{ij}^+ \text{ is nonbasic, } i \in \{1, \dots, n\}, j \in \{n+1, \dots, n+m\}\}$$

$$S_7 = \{x_{ij} : x_{ij}^+ \text{ and } x_{ij}^- \text{ are nonbasic, } i \in \{1, \dots, n\}, j \in \{n+1, \dots, n+m\}\}.$$

For each combination of $i, j \in \{1, \dots, n\}$ such that $i > j$, there are

$$\binom{n}{2} = \frac{n(n-1)}{2} \text{ } x_{ij} \text{ variables. Hence, the set } \{S_2 \cup S_3 \cup S_4\} \text{ has } \frac{n(n-1)}{2} \text{ } x_{ij} \text{'s.}$$

Similarly, for each combination of $i \in \{1, \dots, n\}$ and $j \in \{n+1, \dots, n+m\}$

there is one $x_{ij} \in \{S_5 \cup S_6 \cup S_7\}$. Hence the set $\{S_5 \cup S_6 \cup S_7\}$ has $m \cdot n$ x_{ij} 's.

Hence if S_4 has p x_{ij} 's and S_7 has q x_{ij} 's, then the total number of basic variables are:

(i) The Set S_1 has n basic variables.

(ii) The Set $\{S_2 \cup S_3\}$ has $\frac{n(n-1)}{2} - p$ basic variables.

(iii) The Set $\{S_5 \cup S_6\}$ has $m \cdot n - q$ basic variables.

Adding the above, we get the total number of basic variables is equal to

$$n + \frac{n(n-1)}{2} + m \cdot n - (p+q), \text{ which must be equal to } m \cdot n + \frac{n(n-1)}{2} \text{ by the}$$

last section.

$$\text{Hence, } \underline{p + q = n}.$$

We will be taking advantage of this in Section 3.3.1.

3.2.8 Impact of Degeneracy

As discussed in Section 1.3.1, degeneracy has posed as a stumbling block in the development of an efficient primal based simplex solution to Problem P_x (35). Degeneracy arises when more than one new facility are located at the same x coordinate in any basic feasible solution. Let us investigate the number of degenerate bases that can represent the same solution.

Let $t(\leq n)$ new facilities be located at $x_a (a \in \{n+1, \dots, n+m\})$. Let $S_t = \{\text{Set of indices of these } t \text{ sources}\}$, $S_f = \{\text{Set of indices for the } (m-1) \text{ destinations excluding "a"}\}$, and $S_v = \{\text{Set of indices } i \text{ for the } (n-t) \text{ sources with } x_i \neq x_a\}$.

(i) From Lemma 3.2.6, the number of basic variables involving only the $(n-t)$ sources $\in S_v$ are $(n-t)m + \frac{(n-t)(n-t-1)}{2}$.

(ii) For $i \in S_t$ and $j \in S_f$, either $x_{ij}^+ > 0$ ($x_i > x_j$) or $x_{ij}^- > 0$ ($x_i < x_j$). This gives us $t(m-1)$ basic variables.

(iii) For each combination of $i \in S_t$ and $j \in S_v$ such that $i > j$, either x_{ij}^+ or $x_{ij}^- > 0$ (i.e. either $x_i > x_j$ or $x_i < x_j$).

Also, for each combination of $i \in S_t$ and $j \in S_v$ such that $i < j$, either x_{ji}^+ or $x_{ji}^- > 0$ (i.e. either $x_j > x_i$ or $x_j < x_i$). This gives us $t(n-t)$ basic variables.

(iv) Lastly, the t sources with $i \in S_t$ will have x_i basic. (Note that, the basic x_i with $i \in S_v$ are accounted for in (i) above). There are hence t more basic variables.

We have hence accounted for $\{\frac{n(n-1)}{2} + n \cdot m\} - \frac{t(t-1)}{2}$ basic variables which leaves us with $\frac{t(t-1)}{2}$ basic variables to account for. By examining

the possible basic variables as listed in the last section, we see that we have not considered as yet the x_{ij} variables with $i \in S_t$ and $j \in S_t \cup \{a\}$, $i \neq j$. For $i \in S_t$ and $j = a$ we get t x_{ij} 's and for each combination of $i, j \in S_t$ ($i > j$) we get $\binom{t}{2} = \frac{t(t-1)}{2}$ x_{ij} 's. For these x_{ij} 's both x_{ij}^+ and x_{ij}^- are zero.

We can hence select arbitrarily $\frac{t(t-1)}{2}$ variables from $t + \frac{t(t-1)}{2}$ variables and call them basic. Also, for each x_{ij} we select, we may make either x_{ij}^+ or x_{ij}^- basic. Hence, the number of degenerate bases possible are

$$2^{\frac{t(t-1)}{2}} \cdot \binom{t + \frac{t(t-1)}{2}}{\frac{t(t-1)}{2}}.$$

e.g., when $t = 5$, there are $2^{10} \binom{15}{10}$ possible bases, a staggering figure. Even with $t = 2$, 6 alternate bases are possible. Besides, degeneracy at some other location would magnify this figure. We will now attempt to overcome the dilemma of choosing a suitable basis.

Suppose we have a basic feasible solution to Problem P_x of section 3.2.3 and let us focus our attention on a new facility z with $x_z = x_a$. ($z \in \{1, \dots, n\}$ and $a \in \{n+1, \dots, n+m\}$). As mentioned in Lemma 3.2.5, letting $x_z = x_{a+1}$ corresponds to a simplex pivot. We wish to investigate if this will give a strict improvement in the objective function value. In order to see this, we define a basis in the next section in such a way that if the reduced cost coefficient $(z_{ij} - \bar{c}_{ij})$ of x_{za}^+ is positive then a strict decrease in the objective function is obtained by making $x_z = x_{a+1}$. This will be demonstrated by proving that in such a

case, when x_{za}^+ enters the basis, a non-degenerate pivot occurs.

Similarly, we will define another alternate basis (if necessary) to investigate any potential improvements in making $x_z = x_{a-1}$. This will be done for each new facility. We will then develop a sufficient condition for a strict improvement in the objective function value when moving a single new facility either to the left or the right.

It is necessary to emphasize here that in the absence of degeneracy, a unique basis exists. Hence the need for selecting a suitable basis arises only in the presence of degeneracy.

3.3 Consideration of Single Movements

3.3.1 Definition of a Suitable Basis for Testing the Effectiveness of Moving a Single New Facility "To the Right"

Let the new facility z under consideration have $x_z = x_a$, $z \in \{1, \dots, n\}$ and $a \in \{n+1, \dots, n+m\}$.

In Section 3.2.7 we had indicated that given a basic feasible solution, we could classify the basic variables into the sets S_1 through S_7 which we had defined. For examining the potential improvement of the objective function value by making $x_z = x_{a+1}$, we specify the elements of these sets below:

- (a) All the elements x_1, \dots, x_n of S_1 are basic
- (b) The elements of S_2 and S_3 are determined from the following criteria:
If for $i \in \{1, \dots, n\}$ and $j \in \{i+1, \dots, n\}$,

if $x_i > x_j$, x_{ij}^+ is basic

if $x_i < x_j$, x_{ij}^- is basic

if $x_i = x_j$ and if $i \neq z$, $j \neq z$, x_{ij}^- is basic

if $x_i = x_j$ and if $i = z$ or if $j = z$, then,

if $i = z$, for $j \in \{z+1, \dots, n\}$ if $x_z = x_j$ then x_{zj}^+ is basic, and if

$j = z$, for $i \in \{1, \dots, z-1\}$ if $x_z = x_i$, then x_{iz}^- is basic.

(c) Since for each $i \in \{1, \dots, n\}$ and $j \in \{i+1, \dots, n\}$, either x_{ij}^+ or x_{ij}^- has been defined basic in (b) above, Set S_4 is empty here. i.e. $p = 0$.

(d) The elements of S_5 are determined from the criteria that if for $i \in \{1, \dots, n\}$, $j \in \{n+1, \dots, n+m\}$ if $x_i > x_j$, x_{ij}^+ is basic.

(e) Set S_6 is formed from the criteria that if for $i \in \{1, \dots, n\}$, $j \in \{n+1, \dots, n+m\}$, if $x_i < x_j$, x_{ij}^- is basic.

(f) Since $p = 0$ and $p+q = n$ according to section 3.2.7, $q = n$ i.e. Set S_7 has n elements. These n x_{ij} 's arise from having both x_{ij}^+ and x_{ij}^- non-basic for $i \in \{1, \dots, n\}$, $j \in \{n+1, \dots, n+m\}$ such that $x_i = x_j$.

Hence, S_1 has n basic variables. Further, because $p = 0$, then by Section 3.2.7, $\{S_2 \cup S_3\}$ has $\frac{n(n-1)}{2}$ basic variables. Also since $q = n$, then by Section 3.2.7, $\{S_5 \cup S_6\}$ has $(m \cdot n - n)$ basic variables. This gives us a total of $m \cdot n + \frac{n(n-1)}{2}$ basic variables. Let B_{za^+} be the matrix defined by the columns of the above basic variables with the rows arranged such that the constraints with the basic x_{ij}^+ appear first, those with the basic x_{ij}^- appear next and those involving x_{ij} with $i = 1, \dots, n$ and $j \in \{n+1, \dots, n+m\}$ such that $x_i = x_j$ appear last. We then have,

$$B_{za^+} = \begin{bmatrix} \begin{matrix} x_{ij}^+ (i \in \{1, \dots, n\}, \\ j \in \{i+1, \dots, n\}) \end{matrix} & \begin{matrix} x_{ij}^- (i \in \{1, \dots, n\}, \\ j \in \{i+1, \dots, n\}) \end{matrix} & x_1, \dots, x_n \end{bmatrix}$$

$$= \begin{bmatrix} \hline -I & 0 & A^+ \\ \hline 0 & I & A^- \\ \hline 0 & 0 & I \\ \hline \end{bmatrix}$$

Here, corresponding to the constraints involving the basic x_{ij}^+, A^+ has

+1 under x_i and -1 under x_j if $j \leq n$

+1 under x_i and 0 elsewhere if $j > n$.

Similarly, corresponding to the constraints involving the basic x_{ij}^-, A^- has

+1 under x_i and -1 under x_j if $j \leq n$

+1 under x_i and zeroes elsewhere otherwise.

Lemma 3.3.2. Let B_{za^+} be as defined above. Then

(i) $B_{za^+}^{-1}$ exists.

(ii) The reduced cost coefficient for x_{za}^+ is given by

$$r_z = \sum_{j/x_j > x_z} (w_{zj} + w_{jz}) - \sum_{j/x_j < x_z} (w_{zj} + w_{jz}).$$

(iii) If $r_z > 0$, then a strict improvement in objective function value is obtained by letting $x_z = x_{a+1}$.

Proof:

(i) Clearly the determinant of the Basis Matrix B_{za^+} is nonzero (in fact, it is $= \pm 1$). This implies that the columns are linearly independent and since the rank of this is $m \cdot n + \frac{n(n-1)}{2}$, by Lemma 3.2.6, this is a legitimate basis.

Also,

$$B_{za^+}^{-1} = \begin{array}{c|cc} & \text{"Basic } x_{ij}^+ \text{"} & \text{"Basic } x_{ij}^- \text{"} & \text{"}x_i \text{"} \\ \hline & -I & 0 & A^+ \\ \hline & 0 & I & -A^- \\ \hline & 0 & 0 & I \\ \hline \end{array}$$

(ii) The cost coefficient vector of these basic variables from the objective function is

$$c_B = \begin{bmatrix} \text{"}x_{ij}^+\text{"} & \text{"}x_{ij}^-\text{"} & \text{"}x_i\text{"} \\ \{2w_{ij}\} & \{0\} & \{-w_i\} \end{bmatrix}.$$

With the rows arranged in the same order as for B_{za^+} , the column of coefficients for x_{za}^+ is

$$a_{za^+} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ -1 \\ \vdots \\ 0 \end{bmatrix} \equiv \begin{array}{l} \text{column of zeroes having as many elements} \\ \text{as the number of basic } x_{ij}^+ \\ \\ \text{column of zeroes having as many elements} \\ \text{as the number of basic } x_{ij}^- \\ \\ \text{corresponding to } x_1 \\ \vdots \\ \text{corresponding to } x_z \\ \vdots \\ \text{corresponding to } x_n \end{array}$$

We now define Sets

$$S_a^+ = \{\text{set of indices } j \in \{z+1, \dots, n+m\}, j \neq a \mid x_z \geq x_j\}$$

$$S_a^- = \{\text{set of indices } j \in \{z+1, \dots, n+m\} \mid x_z < x_j\}$$

$$S_b^+ = \{\text{set of indices } i \in \{1, \dots, z-1\} \mid x_z \geq x_i\}$$

$$S_b^- = \{\text{set of indices } i \in \{1, \dots, z-1\} \mid x_z < x_i\}.$$

Hence, the updated column of coefficients for $x_{za}^+ \equiv c_{za^+} = B_{za^+}^{-1} a_{za^+}$

$$\therefore c_{za^+} = \begin{array}{|l} -1 & \text{for each basic } x_{zj}^+ \quad (j \in S_a^+) \\ +1 & \text{for each basic } x_{iz}^+ \quad (i \in S_b^-) \\ 0 & \text{otherwise} \\ \hline -1 & \text{for each basic } x_{iz}^- \quad (i \in S_b^+) \\ +1 & \text{for each basic } x_{zj}^- \quad (j \in S_a^-) \\ 0 & \text{otherwise} \\ \hline -1 & \text{corresponding to } x_z \\ 0 & \text{otherwise.} \end{array}$$

$$\therefore z_{za^+} - \bar{c}_{za^+} = r_z = c_B [B_{az^+}^{-1} \cdot a_{za^+}] - \bar{c}_{za^+} = c_B c_{za^+} - \bar{c}_{za^+}$$

$$= [-2 \sum_{j \in S_a^+} w_{zj} + 2 \sum_{i \in S_b^-} w_{iz} + w_z] = 2w_{za}$$

$$= -2 \sum_{j \in S_a^+} w_{zj} + 2 \sum_{i \in S_b^-} w_{iz} - 2w_{za} + \left[\sum_{j \in S_a^+ \cup S_a^- \cup \{a\}} w_{zj} - \sum_{i \in S_b^+ \cup S_b^-} w_{iz} \right]$$

(by definition of w_z)

$$= - \sum_{j \in S_a^+} w_{zj} + \sum_{j \in S_a^-} w_{zj} + \sum_{i \in S_b^-} w_{iz} - \sum_{i \in S_b^+} w_{iz} - w_{za}$$

$$= - \left[\sum_{j \in S_a^+ \cup \{a\}} w_{zj} + \sum_{i \in S_b^+} w_{iz} \right] + \left[\sum_{j \in S_a^-} w_{zj} + \sum_{i \in S_b^-} w_{iz} \right].$$

$$\therefore r_z = \sum_{j/x_j > x_z} (w_{zj} + w_{jz}) - \sum_{j/x_j < x_z} (w_{zj} + w_{jz}).$$

(iii) Suppose $r_z > 0$ and we enter it in the basis. The candidates for leaving basic variables are those which have a ± 1 in the corresponding position in the c_{za^+} column. These are

$$x_{iz}^+ \quad (i \in S_b^-)$$

and
$$x_{zj}^- \quad (j \in S_a^-)$$

But

$$\text{For } i \in S_b^-, x_{iz}^+ = (x_i - x_z) > 0 \quad \text{and}$$

$$\text{For } j \in S_a^-, x_{zj}^- = x_j - x_z > 0.$$

Also, $\min. x_{zj}^- = x_{a+1} - x_a$ and $\min. x_{iz}^+ \geq x_{a+1} - x_a$ where the equality holds if $x_i = x_{a+1}$ for $i \in S_b^-$. In either case, the leaving basic variable is $x_{z,a+1}^-$ and x_{za}^+ enters at a positive level of $(x_{a+1} - x_a)$ i.e. it has moved to an adjacent position on the right. The decrease in the objective function value is $r_z(x_{a+1} - x_a) > 0$. Hence, with this definition of a basis, if $r_z > 0$, a strict improvement is possible because of a non-degenerate pivot.

3.3.3 Definition of a Suitable Basis for Testing the Effectiveness of Moving a Single New Facility to the Left

We will work similarly as in the last section here. Let the locations be as specified there. The basic variables are picked as in Section 3.3.1, except the elements of S_2 and S_3 (case (b)) which are:

(b) Sets S_2 and S_3 are formed according to the following criteria:

If for $i \in \{1, \dots, n\}$, $j \in \{i+1, \dots, n\}$ if $x_i > x_j$ then x_{ij}^+ is basic

if $x_i < x_j$ then x_{ij}^- is basic

if $x_i = x_j$ and $i \neq z$, $j \neq z$, x_{ij}^- is basic

For i or $j = z$, consider,

For $j \in \{z+1, \dots, n\}$ if $x_z = x_j$, then x_{zj}^- is basic and

for $i \in \{1, \dots, z-1\}$, if $x_z = x_i$, then x_{iz}^+ is basic.

Again, we have $m \cdot n + \frac{n(n-1)}{2}$ variables which form the basic matrix below:

$$B_{za^-} = \begin{array}{c|cc} & \begin{array}{c} x_{ij}^+, \{i \in \{1, \dots, n\}, \\ j \in \{i+1, \dots, n+m\}\} \end{array} & \begin{array}{c} x_{ij}^-, \{i \in \{1, \dots, n\}, \\ j \in \{i+1, \dots, n+m\}\} \end{array} & x_1, \dots, x_n \\ \hline & -I & 0 & A^+ \\ \hline & 0 & I & A^- \\ \hline & 0 & 0 & I \end{array}$$

where A^+ and A^- have the same form as in the last section.

Lemma 3.3.4. Let B_{za^-} be as defined above.

- (i) $B_{za^-}^{-1}$ exists.
- (ii) The reduced cost coefficient for x_{za^-} is given by

$$\ell_z = \sum_{j/x_z > x_j} (w_{zj} + w_{jz}) - \sum_{j/x_z < x_j} (w_{zj} + w_{jz})$$

- (iii) If $\ell_z > 0$, then a strict improvement in the objective function value is obtained by letting $x_z = x_{a-1}$.

Proof:

- (i) $\text{Det}|B_{za^-}| = \pm 1$. Hence the columns of our Basis matrix are linearly independent. Hence we do have a basis.

The Basis inverse is given by

$$B_{za^-}^{-1} = \begin{array}{c|cc} & x_{ij}^+ & x_{ij}^- & x_i \\ \hline & -I & 0 & A^+ \\ \hline & 0 & I & -A^- \\ \hline & 0 & 0 & I \end{array}$$

- (ii) The basic cost coefficient vector is

$$c_B = \begin{bmatrix} x_{ij}^+ & x_{ij}^- & x_i \\ \{2w_{ij}\} & \{0\} & \{-w_i\} \end{bmatrix}$$

Here again, a_{za^-} is (with rows rearranged as for B_{za^-})

$$a_{za^-} = \begin{bmatrix} 0 & \text{as many zeroes as basic } x_{ij}^+ \\ 0 & \text{as many zeroes as basic } x_{ij}^- \\ 0 & \text{corresponding to } x_1 \\ \vdots & \vdots \\ 1 & \text{corresponding to } x_z \\ \vdots & \vdots \\ 0 & \text{corresponding to } x_n \end{bmatrix}$$

We define the sets

$$S_c^+ = \{\text{set of indices } j \in \{z+1, \dots, n+m\} / x_z > x_j\}$$

$$S_c^- = \{\text{set of indices } j \in \{z+1, \dots, n+m\}, j \neq a / x_z \leq x_j\}$$

$$S_d^+ = \{\text{set of indices } i \in \{1, \dots, z-1\} / x_z > x_i\}$$

$$S_d^- = \{\text{set of indices } i \in \{1, \dots, z-1\} / x_z \leq x_i\}$$

As before, the updated column c_{za^-} of x_{za^-} is given by $c_{za^-} = B_{za^-}^{-1} \cdot a_{za^-}$.

$$\therefore c_{za^-} = \begin{bmatrix} +1 & \text{for each basic } x_{zj}^+ (j \in S_c^+) \\ -1 & \text{for each basic } x_{iz}^+ (i \in S_d^-) \\ 0 & \text{otherwise} \\ +1 & \text{for each basic } x_{iz}^- (i \in S_d^+) \\ -1 & \text{for each basic } x_{zj}^- (j \in S_c^-) \\ 0 & \text{otherwise} \\ +1 & \text{corresponding to } x_z \\ 0 & \text{otherwise} \end{bmatrix}$$

The reduced cost coefficient for $x_{za^-} = \ell_z = (z_{za^-} - \bar{c}_{za^-}) = \bar{c}_B^{-1} \cdot B_{za^-}^{-1} \cdot a_{za^-}$

$$= c_B c_{za^-} = 2 \sum_{j \in S_c^+} w_{zj} - 2 \sum_{i \in S_d^-} w_{iz} - W_z$$

$$= 2 \sum_{j \in S_c^+} w_{zj} - 2 \sum_{i \in S_d^-} w_{iz} - \sum_{j \in S_c^+ \cup S_c^- \cup \{a\}} w_{zj} + \sum_{i \in S_d^+ \cup S_d^-} w_{iz}$$

(using the definition of W_z).

$$= \sum_{j \in S_c^+} w_{zj} - \sum_{i \in S_d^-} w_{iz} - \sum_{j \in S_c^- \cup \{a\}} w_{zj} + \sum_{i \in S_d^+} w_{iz}$$

$$\therefore \ell_z = \sum_{j/x_z > x_j} (w_{zj} + w_{jz}) - \sum_{j/x_z < x_j} (w_{zj} + w_{jz})$$

$$\text{Further, } \ell_z = -r_z - 2 \sum_{j/x_z = x_j} (w_{zj} + w_{jz}).$$

(iii) Suppose $\ell_z > 0$ and we enter it in the basis. The likely candidates for leaving the basis are those which have +1 in the corresponding position in the c_{za^-} column. These are

$$x_{zj}^+ \quad (j \in S_c^+)$$

and $x_{iz}^- \quad (i \in S_d^+)$

and $x_z.$

By a suitable choice of the origin, we can assume that d_j 's are sufficiently large so that x_z will never be pivoted out of the basis.

For $j \in S_c^+$, $x_{zj}^+ = (x_z - x_j) > 0$ and

for $i \in S_d^+$, $x_{iz}^- = (x_z - x_i) > 0.$

Min. x_{zj}^+ for $j \in S_c^+$ is $(x_a - x_{a-1})$ {at $j = a-1$ } and
 min. x_{iz}^- for $i \in S_d^+$ is $\geq (x_a - x_{a-1})$. Hence, x_{za}^- will enter the basis
 at a positive level of $(x_a - x_{a-1})$ and effectively, new facility z will
 have moved to the adjacent position on the left. The objective function
 will have decreased by $\ell_z^+(x_a - x_{a-1}) > 0$. Again, we observe that if
 we find that $\ell_z > 0$, we can simply move the new facility to the adjacent
 position on the left.

If after the above analyses, we find that $r_i \leq 0$, $\ell_i \leq 0$ for all
 $i = 1, \dots, n$, then this implies that moving a single facility either to
 the right or to the left is disadvantageous. Further, if there is no
 degeneracy, the optimum solution is at hand. This follows immediately
 from the following three properties associated with a simplex iteration:
 (These lemmas are given without proof and an interested reader is referred
 to reference (35)).

Lemma 3.3.5. A simplex pivot operation would not involve the movement
 of a subset of new facilities at a given location to a non-adjacent loca-
 tion.

Lemma 3.3.6. Consider two or more mutually exclusive subsets of new
 facilities with each new facility in a given subset being at the same loca-
 tion, but each of the subsets being at a different location. A simplex
 pivot would not involve movement of two or more such subsets of new facili-
 ties.

Lemma 3.3.7. A simplex pivot would not move one subset of new facilities
 (at a given location) to the right and another mutually exclusive subset
 of new facilities (at the same given location) to the left.

3.4 Consideration of Joint Movements

Hereafter, we will refer to new facilities located at the same x coordinate (in Problem P_x) as "degenerate new facilities." In the previous sections we have shown how to detect if moving a single new facility to an adjacent location will result in an improvement of the objective function value. Lemmas 3.3.5, 3.3.6, and 3.3.7 indicate that joint movements of degenerate facilities are also possible. We will investigate this now.

Consider a particular location with say, h degenerate facilities. Let these form set S_h . We are interested in determining whether a subset S_t of S_h , containing t of these h facilities can be moved jointly to the right, say, and give a strict decrease in the objective function value. In Problem P_x (of section 3.2.1), for $i, j \in S_t$, $x_{ij}^+ = x_{ij}^- = 0$ (since $x_i = x_j$), and hence, for considering the joint movement of S_t , we can equivalently let $w_{ij} = 0$ for $i, j \in S_t$.

Let the new facility thus obtained by combining the t facilities be indexed as \bar{t} . We will now treat this as a single new facility and determine its reduced cost coefficient $r_{\bar{t}}$ as in Lemma 3.3.2 (ii). Then by Lemma 3.3.2 (iii), if $r_{\bar{t}} > 0$, then we can move all the t facilities

S_t to the right and obtain an improvement in the objective function value. Since the t new facilities behave as a single new facility \bar{t} , we have,

$$w_{\bar{t}j} = \sum_{i \in S_t} w_{ij} \quad \text{for } j \notin S_t$$

and

$$w_{j\bar{t}} = \sum_{i \in S_t} w_{ji} \quad \text{for } j \notin S_t.$$

Let $x_{\bar{t}} = x_a$ ($a \in \{n+1, \dots, n+m\}$) be the location of S_h . Then Lemma 3.3.2

(ii) gives us

$$\begin{aligned}
 r_{\bar{t}} &= \sum_{j/x_j > x_a} (w_{\bar{t}j} + w_{j\bar{t}}) - \sum_{\substack{j/x_j \leq x_a \\ j \notin S_t}} (w_{\bar{t}j} + w_{j\bar{t}}) \\
 &= \sum_{j/x_j > x_a} \sum_{i \in S_t} (w_{ij} + w_{ji}) - \sum_{\substack{j/x_j \leq x_a \\ j \notin S_t}} \sum_{i \in S_t} (w_{ij} + w_{ji}). \\
 &= \sum_{i \in S_t} \left[\sum_{j/x_j > x_a} (w_{ij} + w_{ji}) - \sum_{j/x_j \leq x_a} (w_{ij} + w_{ji}) + \sum_{j \in S_t} (w_{ij} + w_{ji}) \right] \\
 \therefore r_{\bar{t}} &= \sum_{i \in S_t} \left[r_i + \sum_{j \in S_t} (w_{ij} + w_{ji}) \right] \\
 &= \sum_{i \in S_t} r_i + \sum_{i \in S_t} \left[\sum_{\substack{j < i \\ j \in S_t}} (w_{ij} + w_{ji}) + \sum_{\substack{j > i \\ j \in S_t}} (w_{ij} + w_{ji}) \right] \{ \because w_{ii} = 0 \}
 \end{aligned}$$

But by the definition of w_{ij} for $i, j \in \{1, \dots, n\}$, we know that

$$w_{ij} = 0 \quad \text{if } j < i.$$

$$\therefore r_{\bar{t}} = \sum_{i \in S_t} r_i + \sum_{i \in S_t} \sum_{\substack{j < i \\ j \in S_t}} w_{ji} + \sum_{i \in S_t} \sum_{\substack{j > i \\ j \in S_t}} w_{ij}$$

By merely interchanging i and j we see the last two terms are equal

$$\therefore r_{\bar{t}} = \sum_{i \in S_t} r_i + \sum_{i \in S_t} \sum_{\substack{j > i \\ j \in S_t}} 2w_{ij}$$

For each pair of i and $j \in S_t$ ($i \neq j$) from the last term above, we get $2w_{ij}$ if $i < j$ and $2w_{ji}$ if $i > j$.

$$\therefore \text{Let } a_{ij}^1 \text{ (defined for each pair } i, j \in S_t) = \begin{cases} 2w_{ij} & \text{if } i < j \\ 2w_{ji} & \text{if } i > j \end{cases}$$

Then

$$r_{\overline{T}} = \sum_{i \in S_t} r_i + \sum_{(i,j) \in S_t} a_{ij}^1.$$

3.4.1 Formulation of a Graph-Theory Problem

In the last section, we introduced a means of evaluating whether there is a potential advantage in moving the t new facilities jointly to the right. However, we have yet to determine which of the h degenerate facilities should belong to set S_t . Since a combinatorial approach will be tedious, we will develop an efficient method to solve this problem.

We will formulate the problem of determining S_t as a graph theory problem. By doing this, we will not only solve our own problem but will provide a solution procedure for an interesting graph theory problem.

Consider a graph with each node representing a new facility $\in S_h$. Each node has a weight r_i attached to it. Each node is connected directly to every other node. Let the arc connecting nodes i and j have a weight a_{ij}^1 associated with it.

By a SUBGRAPH S_t , we mean the subset S_t of the nodes $\in S_h$ and all the arcs connecting these nodes $\in S_t$ to each other. By the WEIGHT OF THIS SUBGRAPH, we mean the sum of the weights on the nodes and the arcs associated with it. It is clear that the $r_{\overline{T}}$ of the previous section is precisely this subgraph weight. We are hence interested in a subgraph of maximum weight. If this $r_{\overline{T}}^* > 0$, it is advantageous to move S_t to the

adjacent position on the right. If $r_{\bar{e}}^* \leq 0$, then there is no subset of S_h which can be moved jointly to the right to improve the objective function value. (A similar consideration holds for movements to the left).

Letting $a_{ij} = \frac{1}{2} a'_{ij}$ and renumbering the h facilities $\in S_h$ as $1, \dots, h$, we may formulate this graph theory problem as the following quadratic assignment problem:

$$\begin{aligned} \underline{\text{max.}} \quad & \sum_{i=1}^h x_i r_i + \sum_{i=1}^h \sum_{j=1}^h x_i x_j a_{ij} \\ \underline{\text{subj. to}} \quad & x_i = 0, 1 \quad i=1, \dots, h. \end{aligned}$$

We now define $r = \begin{bmatrix} r_1 \\ \vdots \\ r_h \end{bmatrix}$, $x = \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}$

and

$$A = \begin{bmatrix} 0 & a_{12} & \dots & a_{1h} \\ a_{12} & 0 & \dots & a_{2h} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1h} & a_{2h} & \dots & 0 \end{bmatrix}$$

where $A_{h \times h}$ is a symmetric matrix.

We will now relax the integer constraints $x_i = 0$ or 1 to the form $0 \leq x_i \leq 1$. We will justify this by showing that the solution to the problem with these relaxed constraints satisfies the original integer constraints. The problem may be reformulated as:

PROBLEM P_G

$$\begin{aligned} \underline{\text{max.}} \quad & f(x) = r^t x + x^t A x \\ \underline{\text{subj. to}} \quad & 0 \leq x_i \leq 1 \quad i=1, \dots, h. \end{aligned}$$

We will use Cabot and Francis' Algorithm (7) to solve this problem. However, Cabot and Francis' procedure is designed for a general quadratic programming problem:

$$\begin{aligned} \underline{\text{min.}} \quad & r^t x + x^t A x \\ \underline{\text{subj. to}} \quad & Bx \leq d \\ & x \geq 0. \end{aligned}$$

In our problem (P_G), the constraint set is of a much simpler form so that the solutions to the sub-problems of Cabot and Francis' Algorithm may be trivially obtained. Further, the algorithm calls for the use of Murty's (32) extreme point ranking procedure. The extreme points of our constraint set are easily identifiable and hence an efficient ranking procedure may be developed.

3.4.2 Algorithm CFA for the Solution of Problem P_G

Here, we will describe the steps of Cabot and Francis' Algorithm as specialized for problem P_G :

STEP 1: Solve for $j = 1, \dots, h$

$$\begin{aligned} \underline{\text{max.}} \quad & (a^j)^t x \\ \underline{\text{subj. to}} \quad & 0 \leq x_i \leq 1 \quad (i=1, \dots, h). \end{aligned}$$

where, a^j is the j^{th} column of matrix A .

Here, since $A \geq 0$, the optimum solution to this has the objective function value:

$u_j = (a^j)^t \cdot \underline{1}$ where $\underline{1}$ is a column vector of h elements, all equal to unity. (i.e. $x_i = 1$ for $i = 1, \dots, h$).

STEP 2: Solve problem P_S :

$$\begin{aligned} \underline{\text{max.}} \quad g(x) &= \sum_{j=1}^h (r_j + u_j) x_j = \sum_{j=1}^h c_j x_j \\ \underline{\text{subj. to}} \quad 0 &\leq x_j \leq 1 \quad (j=1, \dots, h). \end{aligned}$$

Let x_o be the optimal solution. Clearly, $x_o = (x_{o1}, \dots, x_{oh})$ such that $x_{oi} = 1$ if $c_i > 0$ and $x_{oi} = 0$ if $c_i \leq 0$. Then, from the definition of u_j , for any $0 \leq x \leq \underline{1}$, $g(x_o) \geq f(x)$, and since $f(x_o)$ is a particular solution value of problem P_G ,

$$\text{lower bound } f_l = f(x_o)$$

$$\text{upper bound } f_u = g(x_o)$$

Current best solution to $P_G = x_o$.

Hence, if f^* is the optimum solution to P_G , $f(x_o) \leq f^* \leq g(x_o)$. Hence, if $g(x_o) \leq 0$, STOP: $S_t = \{\phi\}$

STEP 3:

Generate the next best extreme point solution x_k to Problem P_S .

(A procedure for this is given in the next section). GO TO STEP 4:

STEP 4: Thus far, we have $f_l \leq f^* \leq f_u$. Now, if $g(x_k) < f_l$, STOP \Rightarrow

The current best solution is optimum for P_G . If $g(x_k) \geq f_l$, replace f_u by $g(x_k)$ to get a new upper bound on f^* . (If $f_u \leq 0$, STOP $\Rightarrow S_t = \{\phi\}$).

If $f(x_k) > f_l$, replace f_l by $f(x_k)$ and let x_k be the current best solution

to Problem P_G . Go to STEP 3. If $f(x_k) \leq f_\ell$, leave all quantities unchanged and go to STEP 3.

At the end of this algorithm, if $f^* > 0$ and $x^* = (x_1^*, \dots, x_n^*)$ is optimum for Problem P_G , then if $x_i^* = 0$, $i \notin S_t$ and if $x_i^* = 1$, $i \in S_t$. S_t may now be moved jointly to the right.

3.4.3 Implementation of Murty's Extreme Point Ranking Procedure

Consider Problem P_S and let us introduce slack variables \bar{x}_i in the corresponding constraints to get

$$\begin{aligned} \text{max.} \quad & \sum_{i=1}^h c_i x_i \\ \text{subj. to} \quad & x_i + \bar{x}_i = 1 \\ & x_i, \bar{x}_i \geq 0 \quad i=1, \dots, h. \end{aligned}$$

Let x_o be the optimum solution and let the basic feasible solution corresponding to this have the basic variables $\equiv (x_{o1}, x_{o2}, \dots, x_{oh})$

where $x_{oi} = x_i$ if $x_i = 1$.

$x_{oi} = \bar{x}_i$ if $x_i = 0$.

Consider any basic feasible solution x_k with basic variables (x_{k1}, \dots, x_{kh}) .

Then it is easily seen that the following are true:

(i) The Basis Matrix $B = I_{h \times h}$. Hence, $B^{-1} = I_{h \times h}$.

(ii) If $C_B \equiv$ vector of cost coefficients for the basic variables, then

$$C_B B^{-1} = C_B = (C_{B1}, \dots, C_{Bh}) \text{ where,}$$

$$\begin{aligned} C_{Bi} &= c_i \quad \text{if } x_i = 1 \text{ i.e. } x_{ki} = x_i \\ &= 0 \quad \text{if } x_i = 0 \text{ i.e. } x_{ki} = \bar{x}_i \end{aligned}$$

(iii) The reduced cost coefficient for the nonbasic variable x_{kj}^1 is given by $(z_j - c_j^1) = C_B B^{-1} a_j - c_j^1 = C_B a_j - c_j^1$ where, a_j is the column of x_{kh}^1 in

the starting tableau and c_j^1 is its cost coefficient in the objective function expression.

a_j has 1 in the j^{th} position and zero elsewhere whether

$$x_{kj}^1 = x_j \quad \text{or} \quad x_{kj}^1 = \bar{x}_j.$$

$$\begin{aligned} \text{and } c_j^1 &= c_j & \text{if } x_{kj}^1 &= x_j \\ &= 0 & \text{if } x_{kj}^1 &= \bar{x}_j \end{aligned}$$

$$\begin{aligned} \therefore (z_j - c_j^1) &= c_j & \text{if } x_j &= 1 \text{ i.e. } x_{kj} = x_j & \text{i.e. } x_{kj}^1 &= \bar{x}_j \\ &= -c_j & \text{if } x_j &= 0 \text{ i.e. } x_{kj} = \bar{x}_j & \text{i.e. } x_{kj}^1 &= x_j. \end{aligned}$$

(iv) Since $B^{-1} = I_{h \times h}$ always, the tableau entries remain unchanged.

Hence, if \bar{x}_i is non-basic and enters the basis, we are guaranteed that x_i will be pivoted out. Hence, every basic feasible solution has either \bar{x}_i or $x_i = 1$ and the other = 0 for $i=1, \dots, h$.

(v) Because of (iv), the RMS is a column of 1's. Hence, we never have degeneracy. This makes Murty's Extreme Point Ranking Procedure extremely efficient (32).

(vi) From (iv) above, it is clear that each basic feasible solution has exactly h adjacent basic feasible solutions. For each component i of the current solution x_k ($i=1, \dots, h$), if it is 1 we can flip it to 0 and vice-versa to get an adjacent basis. In other words, for each component i , if x_i is basic, an adjacent solution would have \bar{x}_i basic and vice versa.

We will now summarize the ranking and enumeration procedure:

Initialization Step: Let x_0 be the optimum solution to Problem P_s as found earlier. Generate the h adjacent points $(x_{0i}, i=1, \dots, h)$ to x_0 by (vi) above and store x_{0i} and $g(x_{0i})$ if $g(x_{0i}) > 0$.

Suppose now that we have a list of extreme points x_e with their corresponding values $g(x_e)$, and suppose that Algorithm CFA has transferred control here to determine the next best extreme point to Problem P_s . Also, assume that at this point, the current bounds on f^* are f_ℓ and f_u and the previous best extreme point generated was x .

Step 1: We are now interested in x_a which is adjacent to x and for which $f_\ell \leq g(x_a) \leq f_u$, since otherwise, we will never have an occasion to use x_a . From (iii) above, we get for each x_a the reduced cost coefficients $(z_j - c_j^1)$ which indicate whether the adjacent extreme point we are enumerating is improving or not. (Note that an improving point already belongs to the list). Also, using (vi) above, we may generate the adjacent extreme points we wish to store as:

Let $x = (x_1, \dots, x_h)$. Then for $i=1, \dots, h$

If $x_i = 0$, then if $c_i \leq 0$ and if $g(x) + c_i \geq f_\ell$, then make $x_i = 1$ in x . (If either condition is violated, go to the next component x_{i+1}).

If $x_i = 1$, then if $c_i \geq 0$ and if $g(x) - c_i \geq f_\ell$, obtain the adjacent extreme point by making $x_i = 0$ in x . (If either condition is violated, go to the next component x_{i+1}).

Each adjacent point x_a generated is stored with its value $g(x_a)$. The extreme point x is dropped from the list of extreme points.

Step 2: From the list of extreme points, pick out that extreme point x_k which has the highest value of g from all those listed. If $g(x_k)$ is the same as for the extreme point picked before it, check to see if x_k has been chosen before. If yes, delete x_k from the list and select another. If not, GO TO STEP 4 of ALGORITHM CFA.

3.5 A Complete Algorithm (MFLOC) For the Solution of Problem P_x

Assume that each new facility is placed at the x coordinate location of some existing facility. Also, we will assume that the necessary adjustments have been made for the case where more than one existing facility have the same x coordinate. Let k_i denote the existing facility at whose x coordinate, new facility i is located $\{i \in \{1, \dots, n\}, k_i \in \{n+1, \dots, n+m\}\}$.

ALGORITHM MFLOC:

Step 1: For each new facility, calculate

$$r_i = \sum_{j/x_j > x_{k_i}} (w_{ij} + w_{ji}) - \sum_{j/x_j < x_{k_i}} (w_{ij} + w_{ji})$$

and

for $i=1, \dots, n$.

$$\lambda_i = -r_i - 2 \sum_{j/x_j = x_{k_i}} (w_{ij} + w_{ji})$$

Step 2: If all $r_i \leq 0$, GO TO STEP 3.

If any $r_i > 0$, GO TO STEP 4.

Step 3: If all $\lambda_i \leq 0$, GO TO STEP 6.

If any $\lambda_i > 0$, GO TO STEP 5.

Step 4: Let $r_t = \max\{r_i\}$. Move new facility t to the adjacent position on its right and update:

$$x_t = x_{(k_t+1)}$$

$$\text{Replace } r_t \text{ by } r_t - 2 \cdot \sum_{j/x_j = x_{(k_t+1)}} (w_{tj} + w_{jt}).$$

Replace ℓ_t by $\ell_t + 2 \cdot \sum_{j/x_j=x_{k_t}} (w_{tj} + w_{jt})$.

For all $i \in \{1, \dots, n\}$ such that $x_i = x_{k_t}$, replace r_i by $r_i + 2(w_{it} + w_{ti})$

For all $i \in \{1, \dots, n\}$ such that $x_i = x_{(k_t+1)}$, replace ℓ_i by $\ell_i - 2(w_{it} + w_{ti})$

Now replace k_t by $k_t + 1$

GO TO STEP 2.

Step 5: Let $\ell_t = \max\{\ell_i\}$. Move new facility t to the adjacent position on its left and update:

$$x_t = x_{k_t-1}.$$

Replace r_t by $r_t + 2 \cdot \sum_{j/x_j=x_{k_t}} (w_{tj} + w_{jt})$

Replace ℓ_t by $\ell_t - 2 \cdot \sum_{j/x_j=x_{(k_t-1)}} (w_{tj} + w_{jt})$

For all $i \in \{1, \dots, n\}$ such that $x_i = x_{(k_t-1)}$, replace r_i by $r_i - 2(w_{it} + w_{ti})$

For all $i \in \{1, \dots, n\}$ such that $x_i = x_{k_t}$, replace ℓ_i by $\ell_i + 2(w_{it} + w_{ti})$

Now replace k_t by $k_t - 1$.

GO TO STEP 2.

Step 6: If each new facility has a distinctly different x coordinate,
STOP - the optimum solution is at hand.

Step 7: Let R be the rightmost position at which degeneracy exists.
Select the leftmost position at which more than one new facilities
co-exist.

Step 8: GO TO ALGORITHM CFA for joint movements to the right.

If $S_t = \phi$, GO TO STEP 9

If $S_t \neq \phi$, move S_t to the adjacent position

on the right. Update all quantities using formulae in step 4 above repetitively, moving each S_t one at a time while computing. GO TO STEP 9.

Step 9: Pick the next position at the right where degeneracy exists.

If incremented value $> R$, GO TO STEP 10. If such a position exists, GO TO STEP 8 after picking up that position.

Step 10. Let L be the leftmost position at which more than one new facilities co-exist. Select the rightmost position of degeneracy. GO TO STEP 11.

Step 11. GO TO ALGORITHM D for joint movements to the left.

If $S_t = \phi$ GO TO STEP 12.

If $S_t \neq \phi$, move S_t to the adjacent position on its left and update according to the method indicated in Step 8.

GO TO STEP 12.

Step 12: Pick the next position of degeneracy on the left. If incremented value $< L$ then,

If no joint movements have been effected either to the left or right during the last pass through steps 7-12, STOP - OPTIMUM SOLUTION IS AT HAND.

If any joint movements have been effected, GO TO STEP 2.

If incremented value $\geq L$ i.e. if the next point of degeneracy exists, pick it up and GO TO STEP 11.

3.6 Illustrative Examples

3.6.1 Illustrative Example 1

For this problem, we will use the counter example proposed by Rao

(35) for the Pritsker and Ghare (34) solution procedure.

Let $n=5$, $m=4$, $x_6 \equiv d_1 = 1 = x_7 \equiv d_2$, $x_8 = d_3 = 2$, $x_9 = d_4 = 3$.

Let the original allocation matrix $[\{w_{ij}^1\}]$ be:

	1	2	3	4	5	6	7	8	9
1	0	1	1	1	1	5	5	0	0
2	1	0	8	1	0	1	3	1	4
3	1	12	0	0	0	3	1	1	4
4	1	0	0	0	30	2	2	5	4
5	1	0	0	10	0	0	4	5	4

Solution: In order to conform to our restriction of

$x_{n+1} < x_{n+2} < \dots < x_{n+m}$, we first combine existing facilities 6 and 7 and call them jointly as 6 and we rename 8 and 9 as 7 and 8 respectively. We now have,

$$x_6 = 1 < x_7 = 2 < x_8 = 3$$

The new $[\{w_{ij}^1\}]$ matrix is displayed partially:

	6	7	8
1	10	0	0
2	4	1	4
3	4	1	4
4	4	5	4
5	4	5	4

We now convert this into the $[w_{ij}]$ format we require to work with:

$$[\{w_{ij}\}] = \begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 0 & 2 & 2 & 2 & 2 & 10 & 0 & 0 \\ 2 & 0 & 0 & 20 & 1 & 0 & 4 & 1 & 4 \\ 3 & 0 & 0 & 0 & 0 & 0 & 4 & 1 & 4 \\ 4 & 0 & 0 & 0 & 0 & 40 & 4 & 5 & 4 \\ 5 & 0 & 0 & 0 & 0 & 0 & 4 & 5 & 4 \end{array}$$

As per Rao's paper, we let $x_1 = 1$, $x_2 = x_3 = x_4 = x_5 = 2$ for a starting solution.

We may then calculate r_i and ℓ_i using the formulae in STEP 1.

New Facility	r_i	ℓ_i
1	-2	-18
2	-24	-20
3	-23	-19
4	-48	-44
5	-47	-43

Since $r_i \leq 0$, $\ell_i \leq 0 \quad \forall i$, no single movements are advantageous. Hence, consider joint movements to the left, say, of the new facilities placed at $x_7 = 2$.

Here,

$$S_h = \{2, 3, 4, 5\},$$

$$A = [\{a_{ij}\}] =$$

	2	3	4	5
2	0	20	1	0
3	20	0	0	0
4	1	0	0	40
5	0	0	40	0

and we may calculate u_i and c_i for algorithm CFA as:

$$\begin{array}{rcccc}
 i & \equiv & 2 & 3 & 4 & 5 \\
 \ell_i & \equiv & -20 & -19 & -44 & -43 \\
 u_i & \equiv & 21 & 20 & 41 & 40 \\
 c_i = \ell_i + u_i & \equiv & 1 & 1 & -3 & -3
 \end{array}$$

Ranked Extreme points x of P_s	$g(x)$	$f(x)$	Bounds on f^*
$x_0 = (1,1,0,0)$	2	1	$[2,1]$

We now generate adjacent points for x_0 and pick the best as the next best. This is x_1 :

$x_1 = (0,1,0,0)$	1	-18	$[1,1] \Rightarrow \text{STOP}$
-------------------	---	-----	---------------------------------

Since $f^* = 1$ and x_0 gives $f(x_0) = 1$, x_0 is optimum

x	Adjacent extreme points for which $g(x) > 0$	Value $g(x)$ is in parenthesis
$x_0 = (1,1,0,0)$	0100 (1) and 1000 (1)	

Hence, $S_t = \{2,3\}$. We hence move new facilities 2 and 3 jointly to the left. Again, $r_i \leq 0$, $\ell_i \leq 0 \quad \forall_i$ and it can easily be checked that no joint movements are possible. Hence the optimal solution is $x_1 = x_2 = x_3 = 1$ and $x_4 = x_5 = 2$. The objective function value is 39.

3.6.2 Illustrative Example 2

Let us use the w_{ij} matrix of the last example but let the starting solution be: $x_1 = 2$, $x_2 = x_3 = x_4 = x_5 = 1$.

Then using the formulae in step 1, we have,

New facility i	r_i	l_i
1	-18	18
2	-18	-32
3	-17	-31
4	-34	-56
5	-33	-55

Here, $l_i > 0$ and we can move 1 to the left. This makes $x_1 = 1$ and the objective function decreases by 18 from 62 to 44. The updated values are: (with $x_1 = x_2 = x_3 = x_4 = x_5 = 1$)

New facility i	r_i	l_i
1	-18	-18
2	-22	-32
3	-21	-31
4	-38	-56
5	-37	-55

Hence, no single movements are possible. We hence have to consider joint movements of $S_h = \{1, 2, 3, 4, 5\}$ to the right.

Ranked Extreme points (x) of P_S	$g(x)$	$f(x)$	Bounds on f^*
(0, 1, 1, 1, 1)	12	4	[12, 4]
(0, 0, 1, 1, 1)	11	-16	[11, 4]
(0, 1, 0, 1, 1)	11	-15	[11, 4]
(0, 0, 0, 1, 1)	10	5	[10, 5]
(0, 1, 1, 0, 1)	7	-40	[7, 5]
(0, 1, 1, 1, 0)	7	-39	[7, 5]
(0, 0, 1, 0, 1)	6	-58	[6, 5]
(0, 0, 1, 1, 0)	6	-59	[6, 5]
(0, 1, 0, 0, 1)	6	-59	[6, 5]
(0, 1, 0, 1, 0)	6	-58	[6, 5]
(0, 0, 0, 0, 1)	5	-36	[5, 5] \Rightarrow STOP.

These extreme points above were generated each time an extreme point was picked up based on its $g(x)$ value to calculate the bounds. Extreme points with $g(x)$ lying outside the current bounds were dropped (not stored). The generation procedure proceeded as shown below.

The maximum value of $f(x) = 5$. The extreme point which gave this value was $(0,0,0,1,1)$ from above. $\therefore S_t = \{4,5\}$. We therefore move S_t to the right and improve the objective function by 5 i.e. it decreases from 44 to 39.

The present solution is $x_1 = x_2 = x_3 = 1, x_4 = x_5 = 2$. Further checks show that this is optimal.

Extreme Point Selected for Bound Calculations	Current Bounds	Adjacent extreme points with Values ($g(x)$) lying in the current bounds
01111	[12,4]	00111(11), 01011(11), 01101 (7), 01110 (7)
00111	[11,4]	00011(10), 00101(6), 00110(6)
01011	[11,4]	00011(10), 01001(6), 01010(6)
00011	[10,5]	00001 (5), 00010(5)
The other point with $g(x)=10$ being the same is not picked.		
01101	[7,5]	00101(6), 01001(6)
01110	[7,5]	00110(6), 01010(6)
00101	[6,5]	00001(5)
00110	[6,5]	00010(5)
01001	[6,5]	00001(5)
01010	[6,5]	00010(5)
00001	[5,5]	-

The above procedure was coded and run on a Cyber 74 machine. The computational times show that it is extremely efficient. These are reported in Chapter VI where they are compared with the results of the Pritsker and Ghare procedure.

CHAPTER IV

THE CUTTING PLANE AND FEASIBLE POINT ALGORITHMS

4.1 Introduction

In this chapter, we will introduce the concept of cutting planes to help us in our search for the global optimum. Cutting planes have been frequently used in integer and non-linear programming, the main objective being to reduce the feasible region to as great an extent as possible with each cut. However, often, these cutting planes destroy the problem structure. In our case, we are strongly concerned with preserving the important structure of the problem even after the cutting planes are introduced.

We will first outline the cutting plane technique developed by Vaish (41) for the general Bilinear Programming Problem. (The basis of this is the theory of polaroids as presented in (6)). We will then extend this concept to develop a technique which produces even deeper cuts.

An algorithm will then be developed to determine a feasible point to the system of cuts. We will then use the cutting plane and the feasible point algorithms iteratively until the feasible region is exhausted.

4.2 The Cutting Plane Algorithm for Bilinear Programming Problems

We will briefly discuss here the concepts introduced by Vaish (41) for generating cutting planes.

4.2.1 Definition: Let $A \subseteq E_m$. Then, given a function $f: E_n \times E_m \rightarrow E_1$, and a scalar k , the generalized polar of A is defined as

$$A^0(k) = \{x \in E_n: f(x, y) \leq k \quad \forall y \in A\}.$$

Relative to our function $\phi(z, u) = d^t u + z^t D u$, we will define our polar sets as

$$U^0(k) = \{z: \phi(z, u) \geq k \quad \forall u \in U\}.$$

Then if $\phi(\bar{z}, \bar{u}) = k$, we may say that (\bar{z}, \bar{u}) is the true minimum in the region $[U^0(k) \times U] \cap [Z \times U]$.

Using $U^0(k)$, suitable cutting planes can be generated which will preserve the structure of our problem since they involve only the variables $z \in Z$. Moreover, the choice of introducing the cuts in the set Z makes the procedure efficient and easily implementable.

Let (\bar{z}, \bar{u}) be the pseudo-global minimum we determined in Chapter II. Consider the polyhedral cone C with its vertex at \bar{z} and whose r extreme rays are given by

$\xi^j = \{z: z = \bar{z} - \bar{e}^j \lambda_j, \lambda_j \geq 0\} \quad \forall j \in J$ where, J is the set of r indices corresponding to the non-basic variables, and \bar{e}^j are the extended non-basic variable columns in the location problem which yielded as its solution the point \bar{z} . (The dimension of \bar{e}^j conforms to that of z).

Now let

$$\begin{aligned} \bar{\lambda}_j &= \max \{ \lambda_j > 0: (\bar{z} - \bar{e}^j \lambda_j) \in U^0(k) \} . \\ &= \max_{\lambda_j > 0} \{ [\min_{u \in U} \phi(\bar{z} - \bar{e}^j \lambda_j, u)] \geq k \} . \end{aligned}$$

If (\bar{z}, \bar{u}) is a pseudo global minimum, $\bar{\lambda}_j \in (0, \infty]$ and $\sum_{j \in J} z_j / \bar{\lambda}_j \geq 1$ is a valid cutting plane (37). (Here, z_j is the j^{th} non-basic variable, $j \in J$. Also, by a valid cutting plane, we mean one which deletes \bar{z} but no point $\hat{z} \in Z$ such that $\phi(\hat{z}, u) < k$ for some $u \in U$).

It may be noted that $\bar{\lambda}_j$ can be found by solving the following parametric problem:

Problem PAR1: Find $\bar{\lambda}_j = \max\{\lambda_j > 0: \psi(\lambda_j) \geq k\}$

where $\psi(\lambda_j) = \min_{u \in U} \{d^t u + (\bar{z} - e^j \lambda_j)^t D u\}$.

4.3 The Negative Edge Extension Method

We will now develop a procedure we will call the "Negative Edge Extension Method" to generate even deeper cuts than those developed above. Consider the j^{th} non-basic variable z_j for which $\bar{\lambda}_j = \infty$ above. We will now consider a "negative extension" of the ray ξ^j i.e. we let $z = \bar{z} + \lambda_j^! e^j$ ($\lambda_j^! \geq 0$). (It may be noted that the polar set $U^0(k)$ is defined by the faces $d^t u^i + z^t D u^i = k$ where u^i are extreme points of U). We now define,

$$J_1 = \{j: j \in J \text{ and } \xi^j \notin U^0(k)\}$$

and let $J_2 = J - J_1$. (Then for problem PAR1, $\bar{\lambda}_j$ is finite for $j \in J_1$ and is infinite for $j \in J_2$).

Lemma 4.3.1. Let (\bar{z}, \bar{u}) be a pseudo-global minimum and let k be the current best value of the objective function. If the set J_1 (defined above) is empty, the current best solution is the global optimum to problem GRLAP.

Proof: Let C be the cone with vertex at \bar{z} and with generators ξ^j . Then,

$$C = \{z: \bar{z} - \sum_{j \in J} \bar{e}^j z_j, z_j \geq 0\}.$$

Since J_1 is empty, this implies that $\xi^j \in U^0(k)$ for each $j \in J$ which means that $C \subseteq U^0(k)$. But the set $Z \subseteq C$. Hence, $Z \subseteq U^0(k)$. By the definition of $U^0(k)$, the solution with objective function value equal to k is optimal for the proposed problem GRLAP.

Hence we note that if $J_1 = \{\phi\}$, the problem is solved and we need not generate any further cuts. If, however, $J_1 \neq \{\phi\}$, then for generating a cut by the negative edge extension, we define $\bar{\lambda}_j$ as:

$$\bar{\lambda}_j = \begin{cases} \max. \{0 < \lambda_j < \infty: \phi(\bar{z} - \bar{e}^j \lambda_j, u) \geq k \ \forall u \in U\} & \text{if } \xi^j \notin U^0(k) \text{ i.e. } j \in J_1 \\ \max. \{0 < \lambda_j \leq \infty; \phi(\bar{z} + \bar{e}^j \lambda_j, u) \geq k \text{ for some } u \in U\} & \text{if } \xi^j \in U^0(k) \\ & \text{i.e. } j \in J_2. \end{cases}$$

Hence, if $\xi_j \in U^0(k)$ to determine $\bar{\lambda}_j$, we have to find the maximum value of λ_j for which $d^t u^i + (\bar{z} + \lambda_j \bar{e}^j)^t D u^i = k$ for some $u^i \in U$ and that for all $\hat{\lambda}_j > \bar{\lambda}_j$, $d^t u + (\bar{z} + \hat{\lambda}_j \bar{e}^j)^t D u < k \ \forall u \in U$. $\bar{\lambda}_j$ is therefore obtained by solving:

Problem PAR 2:

$$\max_{\lambda_j > 0} [\bar{\psi}(\lambda_j) \geq k]$$

$$\text{where, } \bar{\psi}(\lambda_j) = \max_{u \in U} [d^t + (\bar{z} + \lambda_j \bar{e}^j)^t D] u.$$

$$= -\min_{u \in U} [-d^t - (\bar{z} + \lambda_j \bar{e}^j)^t D] u.$$

Hillier and Lieberman [17] have shown that $\bar{\psi}(\lambda_j)$ is concave. The similarity between Problems PAR 1 and PAR 2 indicates that a common solution

technique can be developed for both of them. We will do so in Section 4.6.

Although we have defined $\bar{\lambda}_j$ to be greater than zero above, we have yet to show that such a $\bar{\lambda}_j$ exists.

Lemma 4.3.2. Let (\bar{z}, \bar{u}) be a pseudo-global minimum. Let the current best objective function value be k . If $J_1 \neq \{\phi\}$, then $\bar{\lambda}_j > 0$ for $j \in J$, where $\bar{\lambda}_j$ and the sets J and J_1 are as previously defined.

Proof: Consider $\bar{\lambda}_j$ for $j \in J_1$. Let $\hat{\lambda}_j > 0$ be such that $\hat{z} = \bar{z} - \bar{e}^j \hat{\lambda}_j$ is an adjacent extreme point of \bar{z} relative to the set Z . Since (\bar{z}, \bar{u}) is a pseudo global minimum and k is the current best solution value, we have,

$$k \leq \phi(\bar{z}, \bar{u}) \leq \phi(\hat{z}, u) \quad \forall u \in U.$$

Since $\psi(\lambda_j) = \min_{u \in U} \phi(\bar{z} - \bar{e}^j \lambda_j, u)$ for $\lambda_j \geq 0$, it follows that

$$k \leq \psi(0) \leq \psi(\hat{\lambda}_j).$$

However, $\psi(\lambda_j)$ is concave [17] and is hence quasi-concave. This means that for all λ_j' such that $0 \leq \lambda_j' \leq \hat{\lambda}_j$, $\psi(\lambda_j') \geq k$. Hence,

$$\bar{\lambda}_j = \max_{\lambda_j} [\psi(\lambda_j) \geq k] \geq \hat{\lambda}_j > 0.$$

Now consider $j \in J_2$. We will first show that $\bar{\psi}(0) > k$ where, as before, $\bar{\psi}(\lambda_j) = \max_{u \in U} \phi(\bar{z} + \bar{e}^j \lambda_j, u)$. Hence, $\bar{\psi}(0) = \max_{u \in U} \phi(\bar{z}, u)$. Also,

$\min_{u \in U} \phi(\bar{z}, u) = \phi(\bar{z}, \bar{u}) \geq k$. Hence, $\bar{\psi}(0) \geq k$. Now suppose that $\bar{\psi}(0) = k$.

This implies that $\min_{u \in U} \phi(\bar{z}, u) = \max_{u \in U} \phi(\bar{z}, u) = k$ i.e. $\phi(\bar{z}, u) = k \quad \forall u \in U$.

Hence, \bar{z} lies on all the faces $d^t u^i + z^t D u^i = k$ of the polyhedral set $U^0(k)$. This implies that $U^0(k)$ is a polyhedral cone with vertex at \bar{z} .

Now suppose that for some $p \in J$, $\xi^p \cap U^0(k) = \bar{z}$. Since Z is bounded, this implies that there exists an extreme point z_p of Z , adjacent to \bar{z} and given by $z_p = \bar{z} - \lambda_p \bar{e}^p$ such that $z_p \notin U^0(k)$. Hence by the definition of $U^0(k)$, $\phi(z_p, u_p) < k$ for some $u_p \in U$. But $\phi(\bar{z}, \bar{u}) \geq k$ and (\bar{z}, \bar{u}) is a pseudo-global minimum with z^p an adjacent extreme point to \bar{z} .

Hence $\phi(z_p, u) \geq k \quad \forall u \in U$, a contradiction. Hence, $\bar{\psi}(0) > k$. But $\bar{\psi}(\lambda_j)$ for $\lambda_j \geq 0$ is concave and is hence quasi-concave. Hence, $\bar{\lambda}_j = \max_{\lambda_j} [\bar{\psi}(\lambda_j) \geq k] > 0$.

Theorem 4.3.3. Let (\bar{z}, \bar{u}) be a psuedo global minimum and let k be the current best value of $\phi(z, u)$, $(z, u) \in Z \times U$

Then if $J_1 \neq \{\phi\}$, $\sum_{j \in J_1} \frac{z_j}{\bar{\lambda}_j} - \sum_{j \in J_2} \frac{z_j}{\bar{\lambda}_j} \geq 1$ is a valid cutting plane, i.e.

(i) $\sum_{j \in J_1} \bar{z}_j / \bar{\lambda}_j - \sum_{j \in J_2} \bar{z}_j / \bar{\lambda}_j < 1$ where \bar{z}_j are the non-basic variables

corresponding to the solution \bar{z} and

(ii) There is no point $\hat{z} \in Z$ such that $\min_{u \in U} \phi(\hat{z}, u) < k$ and $\sum_{j \in J_1} \hat{z}_j / \bar{\lambda}_j - \sum_{j \in J_2} \hat{z}_j / \bar{\lambda}_j < 1$, where \hat{z}_j are the non-basic variables corresponding to

the solution \hat{z} .

Proof: (i): Since $J_1 \neq \{\phi\}$, by Lemma 4.3.2, $\bar{\lambda}_j > 0$ for all $j \in J_1 \cup J_2$.

Since all the non-basic variables \bar{z}_j are zero, the left hand side of

(i) is zero and hence inequality (i) holds.

(ii): Let C be the cone with vertex at \bar{z} and with generators ξ^j . i.e.

$C = \{z = \bar{z} - \sum_{j \in J} \bar{e}^j z_j, z_j \geq 0\}$. Let the cutting plane be defined as

$H = \{z: \sum_{j \in J_1} z_j / \bar{\lambda}_j - \sum_{j \in J_2} z_j / \bar{\lambda}_j = 1\}$ and let the closed half space

H^- be defined as, $H^- = \{z: \sum_{j \in J_1} z_j / \bar{\lambda}_j - \sum_{j \in J_2} z_j / \bar{\lambda}_j \leq 1\}$. Let the set

$$S = C \cap H^-.$$

Consider the set $S_1 = \{z: \phi(z, u) < k \text{ for some } u \in U\}$. Then from the definition of $U^0(k)$, $S_1 \cap U^0(k) = \{\phi\}$. Hence, if we show that $S = (H^- \cap C) \subseteq U^0(k)$, then this means that $S \cap S_1 = \{\phi\}$ i.e. we will have proved (ii). We will hence show that $S \subseteq U^0(k)$.

From the definitions of S , H and $\bar{\lambda}_j (j \in J_1 \cup J_2)$, we can characterize S as a polyhedral set with

$$(1) \text{ Extreme points } z_p = \begin{cases} \bar{z}, & \text{and} \\ (\bar{z} - \bar{\lambda}_j \bar{e}^j) & \text{for } j \in J_1 \end{cases}$$

and (2) Extreme directions

$$e_r = \begin{cases} -\bar{e}^j & \text{for } j \in J_2, \\ [(\bar{z} - \bar{\lambda}_p \bar{e}^p) - (\bar{z} + \bar{\lambda}_q \bar{e}^q)] & \text{for } p \in J_1 \text{ and } q \in J_2 \end{cases}$$

Let $P = \{z_p: z_p \text{ is an extreme point of } S\}$ and let

$R = \{e_r: e_r \text{ is an extreme direction of } S\}$. Then by the well known Representation Theorem, if $\hat{z} \in S$, we can write

$$\hat{z} = \sum_{p \in P} \lambda_p z_p + \sum_{r \in R} \mu_r e_r, \quad \lambda_p \geq 0 \text{ for } p \in P, \quad \sum_{p \in P} \lambda_p = 1, \quad \mu_r \geq 0 \text{ for } r \in R$$

Hence for any fixed $\hat{u} \in U$, we have,

$$\phi(\hat{z}, \hat{u}) = (d^t \hat{u} + \hat{z}^t D \hat{u}) = d^t \hat{u} + \left(\sum_{p \in P} \lambda_p z_p + \sum_{r \in R} \mu_r e_r \right)^t D \hat{u}$$

$$\begin{aligned}
&= (d^t \hat{u} \cdot \sum_{p \in P} \lambda_p) + \left(\sum_{p \in P} \lambda_p z_p \right)^t D \hat{u} + \left(\sum_{r \in R} \mu_r e_r \right)^t D \hat{u} \\
&= \sum_{p \in P} \lambda_p (d^t \hat{u} + z_p^t D \hat{u}) + \sum_{r \in R} \mu_r (e_r^t D \hat{u}) \\
\therefore \phi(\hat{z}, \hat{u}) &= \sum_{p \in P} \lambda_p \phi(z_p, \hat{u}) + \sum_{j \in J_2} \mu_j (-\bar{e}^{-j})^t D \hat{u} + \sum_{\substack{r \in R \\ p \in J_1 \\ q \in J_2}} \mu_r [(\bar{z} - \bar{\lambda}_p \bar{e}^p) - (\bar{z} + \bar{\lambda}_q \bar{e}^q)]^t D \hat{u}
\end{aligned}$$

We will now investigate separately the three terms in this equation.

(a) In the first term, note that by the definition of $\bar{\lambda}_j$ for $j \in J_1$, $z_p \in U^0(k)$. Hence for any $\hat{u} \in U$, $\phi(z_p, \hat{u}) \geq k$ and since $\lambda_p \geq 0$, $\sum_{p \in P} \lambda_p = 1$,

we have, $\sum_{p \in P} \lambda_p \cdot \phi(z_p, \hat{u}) \geq k$.

(b) Let us consider the second term now.

$$\begin{aligned}
\sum_{j \in J_2} \mu_j (-\bar{e}^{-j})^t D \hat{u} &= \sum_{j \in J_2} ([d^t \hat{u} + (\bar{z} - \bar{e}^{-j} \mu_j)^t D \hat{u}] - [d^t \hat{u} + \bar{z}^t D \hat{u}]) \\
&= \sum_{j \in J_2} (\phi(z', \hat{u}) - \phi(\bar{z}, \hat{u})) \text{ where } z' = (\bar{z} - \bar{e}^{-j} \mu_j) \in \xi^j, j \in J_2 \\
&\quad \mu_j \geq 0.
\end{aligned}$$

But if $j \in J_2$, $\xi^j \subset U^0(k)$. Hence for any $\hat{u} \in U$,

$$\phi(z', \hat{u}) \geq k \quad \text{and} \quad \phi(\bar{z}, \hat{u}) \geq k.$$

But for a fixed $\hat{u} \in U$, we showed in Section 2.5.2 that the problem

$\min \phi(z, \hat{u})$ is a linear programming problem. We also know that \bar{z} is an extreme point of Z and $\xi^j \cap Z$ is an edge of Z such that $\bar{z} \in \xi^j \cap Z$.

Hence, if for any $\mu_j \geq 0$, $\phi(z', \hat{u}) < \phi(\bar{z}, \hat{u})$, then $-\bar{e}^{-j}$ is an improving direction and this implies that there exists a $\hat{\mu} > 0$ such that

$\phi(\bar{z} - \bar{e}^j \hat{u}, \hat{u}) = \phi(z^1, \hat{u}) < k$ which contradicts that $\phi(z^1, \hat{u}) \geq k$. Hence $\phi(z^1, \hat{u}) \geq \phi(\bar{z}, \hat{u})$ and hence the second term is non-negative.

(c) Now consider the third term. We may re-write this as

$$\begin{aligned} \sum_{\substack{r \in R \\ p \in J_1 \\ q \in J_2}} \mu_r ([d^t \hat{u} + (\bar{z} - \bar{\lambda}_p \bar{e}^p) t_D \hat{u}] - [d^t \hat{u} + (\bar{z} + \bar{\lambda}_q \bar{e}^q) t_D \hat{u}]) \\ = \sum_{\substack{r \in R \\ p \in J_1 \\ q \in J_2}} \mu_r [\phi(z_p, \hat{u}) - \phi(z_q, \hat{u})] \quad \text{where} \end{aligned}$$

$z_p = \bar{z} - \bar{\lambda}_p \bar{e}^p$, $p \in J_1$ and $z_q = (\bar{z} + \bar{\lambda}_q \bar{e}^q)$, $q \in J_2$. But by definition of $\bar{\lambda}_p$, $p \in J_1$ and $\bar{\lambda}_q$, $q \in J_2$

$$\phi(z_p, \hat{u}) \geq k \quad \text{and} \quad \phi(z_q, \hat{u}) \leq k$$

for any $\hat{u} \in U$. Hence the third term is also non-negative.

Combining (a), (b) and (c) we note that $\phi(\hat{z}, \hat{u}) \geq k$ for any arbitrary $\hat{u} \in U$ and hence $\phi(\hat{z}, u) \geq k \quad \forall u \in U$ i.e. $z \in U^0(k)$. Hence $\hat{z} \in S$ implies that $\hat{z} \in U^0(k)$ i.e. $S \subseteq U^0(k)$ and we have a valid cutting plane.

It may be noted that the negative sign term in the cut accounts for the fact that for $j \in J_2$, the value of the non-basic variable z_j is $-\bar{\lambda}_j$ (< 0) at the point which lies on the cutting plane. Hence at this point, $\frac{-z_j}{\bar{\lambda}_j} = 1$ and $z_t = 0 \quad \forall t \in J, t \neq j$.

Let the feasible region defined by $(H_1^+ \cap Z) \times U$ be $Z^1 \times U$ (where H_1^+ is the closed halfspace of all points feasible to the first cut). We will now attempt to find a point $z \in Z^1$ and starting from this point, we will determine, what we will call a 'weak pseudo global minimum,' in the

region $Z^1 \times U$. From this point, we will generate another cut H_2 and redefine the feasible region as $Z^2 \times U = (H_1^+ \cap H_2^+ \cap Z) \times U$. We will iterate in this fashion until for some $n \geq 1$, $(H_1^+ \cap \dots \cap H_n^+) \cap Z = \{\phi\}$. The problem will then be solved.

In the next few sections we will determine the cardinality of the set J and determine explicitly, the structure of \bar{e}^j . We will then show how $\bar{\lambda}_j$ can be computed for $j \in J_1 \cup J_2$ by solving problems PAR 1 and PAR 2.

4.4 Cardinality of Set J

The total number of variables in problem P_x (or P_y) of Chapter III were shown to be $n + n(n-1) + 2mn$; of these, $\frac{n(n-1)}{2} + nm$ were basic. Hence, we had $\frac{n(n-1)}{2} + mn + n$ non-basic variables.

Hence, the cardinality of J = total number of non-basic variables for a given solution $z \in Z = 2(\frac{n(n-1)}{2} + mn + n) = n(2m + n + 1)$.

In the next section, we will show that the solution of problems PAR 1 and PAR 2 need not be carried out for all z_j , $j \in J$ and hence, the procedure is somewhat simplified.

4.5 Development of \bar{e}^j and Determination of $\psi(\lambda_j)$ and $\bar{\psi}(\lambda_j)$

Here, we will make use of the updated columns we had developed for Problem P_x in Chapter III. We define the following sets of non-basic variables:

$$S_{1x} = \{x_{ij}^- \text{ non-basic corresponding to the basic } x_{ij}^+, i, j \in \{1, \dots, n\}\}$$

$$S_{2x} = \{x_{ij}^+ \text{ non-basic corresponding to the basic } x_{ij}^-, i, j \in \{1, \dots, n\}\}$$

$$S_{3x} = \{x_{ij}^- \text{ non-basic corresponding to the basic } x_{ij}^+, i \in \{1, \dots, n\}, \\ x_i \neq x_j \text{ where } j \in \{n+1, \dots, n+m\}\}$$

$$S_{4x} = \{x_{ij}^+ \text{ non-basic corresponding to basic } x_{ij}^-, i \in \{1, \dots, n\}, \\ j \in \{n+1, \dots, n+m\}, x_i \neq x_j\}$$

$$S_{5x} = \{x_{ij}^+ \text{ and } x_{ij}^- \text{ both non-basic for } i \in \{1, \dots, n\}, j \in \{n+1, \dots, n+m\} \\ \text{such that } x_i = x_j\}.$$

(These sets are based on the choice of our basis, in case degeneracy exists, as in Chapter III).

Similarly, we define the sets S_{iy} ($i = 1, 2, 3, 4, 5$) for the y_{ij} variables in problem P_y . Using the updated columns c_{ij}^\pm of Chapter III, for the non-basic variables $x_{ij}^- \in (S_{1x} \cup S_{3x})$ or $y_{ij}^- \in (S_{1y} \cup S_{3y})$, the extended column \bar{e}^j has

- 1 in position of the basic x_{ij}^+ (or y_{ij}^+)
- 1 in position of x_{ij}^- (or y_{ij}^-)
- 0 elsewhere.

For the non-basic variables

$$x_{ij}^+ \in (S_{2x} \cup S_{4x}) \text{ or } y_{ij}^+ \in (S_{2y} \cup S_{4y})$$

$$\bar{e}^j \text{ has } \begin{array}{ll} -1 & \text{in position of the basic } x_{ij}^- \text{ (or } y_{ij}^-) \\ -1 & \text{in position of } x_{ij}^+ \text{ (or } y_{ij}^+) \\ 0 & \text{elsewhere.} \end{array}$$

Hence, increasing any of these non-basic variables by Δ will result in an increase of the corresponding basic variable (its complement) by Δ , and all the other variables remain unaffected. This means, that the solution in the z space is still the same and the objective function value has not changed. Since,

$$\min_{u \in U} (d^t + \bar{z}^t D)u \geq k \quad \text{and} \quad \max_{u \in U} (d^t + \bar{z}^t D)u \geq k,$$

$$\min_{u \in U} (d^t + (\bar{z} - \bar{e}^j \lambda_j^!)^t D)u \geq k \quad \forall \lambda_j^! \geq 0. \quad \therefore \bar{\lambda}_j = \infty, j \in J_1$$

$$\text{and} \quad \max_{u \in U} (d^t + (\bar{z} + \bar{e}^j \lambda_j^!)^t D)u \geq k \quad \forall \lambda_j^! \geq 0 \quad \therefore \bar{\lambda}_j = \infty, j \in J_2$$

We hence need to solve for only the $4n$ parameters $\bar{\lambda}$ for the non-basic variables $\in S_5$. Based on the definition of our basis as in Chapter III, the following characteristics are evident:

- (i) x_1, \dots, x_n are always basic.
- (ii) for $i \in \{1, \dots, n\}, j \in \{n+1, \dots, n+m\}$, $x_i \neq x_j$, all the alternative bases we defined had the same x_{ij}^+ or x_{ij}^- basic.
- (iii) For $i \in \{1, \dots, n\}, j \in \{n+1, \dots, n+m\}$, $x_i = x_j$, x_{ij}^+ and x_{ij}^- were both defined to be non-basic.
- (iv) Hence, the only variables in the alternative basis which may be different (i.e. the degenerate variables) are x_{ij}^\pm for $i, j \in \{1, \dots, n\}$. However, we are basically interested in the expression $\psi(\lambda_j)$ which has the terms $\sum_{k=1}^K t_k (x_{ij}^+ + x_{ij}^-) u_{ijk}$ for $i, j \in \{1, \dots, n\}$. Hence, no matter which basis we consider i.e. the one with x_{ij}^+ basic or x_{ij}^- basic (if this is possible), the corresponding \bar{e}^j will correctly reflect the variation in $(x_{ij}^+ + x_{ij}^-)$.

Hence, for $z \in \{1, \dots, n\}$, $a, b \in \{n+1, \dots, n+m\}$ such that

$$x_z = x_a, y_z = y_b, \bar{e}^j \text{ for } x_{za}^+ \text{ has } \begin{array}{l} c_{za}^+ \\ -1 \text{ in position of } x_{za}^+ \\ 0 \text{ elsewhere.} \end{array} \text{ as in Chapter III}$$

\bar{e}^j for x_{za}^- has c_{za}^- as in Chapter III
 -1 in position of x_{za}^-
 0 elsewhere

\bar{e}^j for y_{za}^+ and y_{za}^- are similar to this.

Hence to solve either $\min_{u \in U} [(d^t + \bar{z}^t D) - \lambda_j (\bar{e}^j)^t D] u$

or $-\min_{u \in U} [(-d^t - \bar{z}^t D) - \lambda_j' (\bar{e}^j)^t D] u$ for a given λ_j or λ_j' the constant

part of the cost coefficients $(\pm(d^t + \bar{z}^t D))$ are incremented in these transportation problems as follows:

(i) When working with the k^{th} transportation problem for determining $\bar{\lambda}$ for x_{za}^+ , modify the cost coefficients according to:

Increase by $t_k \lambda$ for all destinations $j \in \{1, \dots, n+m\}$ such that $x_j \leq x_a$

Decrease by $t_k \lambda$ for all destinations $j \in \{1, \dots, n+m\}$ such that $x_j > x_a$

For all sources $i \in \{1, \dots, n\}$ wherever the destination is z ,

Increase by $t_k \lambda$ if $x_i \leq x_z = x_a$ and decrease by $t_k \lambda$ if $x_i > x_z = x_a$.

(ii) Similarly, for x_{za}^- , modify the cost coefficients according to:

If destination $j \in \{1, \dots, n+m\}$ and the source is z ,

Increase by $t_k \lambda$ if $x_j \geq x_a$

and Decrease by $t_k \lambda$ if $x_j < x_a$.

If sources are $i \in \{1, \dots, n\}$ and destination is z ,

Increase by $t_k \lambda$ if $x_i \geq x_a$

and Decrease by $t_k \lambda$ if $x_i < x_a$.

An identical approach may be taken for \bar{e}^j for y_{zb}^+ and y_{zb}^- .

An examination of this will reveal that, in the x coordinates for instance, increasing x_{za}^+ by $\Delta \lambda$ ($\Delta \lambda \leq x_{a+1} - x_a$) is equivalent to

moving new facility z a distance $\Delta\lambda$ to the right and vice-versa for a $\Delta\lambda$ increase in x_{za}^- .

4.6 Solution of Problems PAR1 and PAR2

We will provide a solution for problem PAR1 which can be used for problem PAR2 with the slight modifications shown in Section 4.6.2.

In references (41) and (37) this problem is solved using a Bolzano Search Technique. However, we will exploit the special structure of piece-wise linearity of $\psi(\lambda_j)$ and use Newton's method to obtain an even more efficient convergent method. In Chapter VI we demonstrate the superiority of this technique. (It may be noted here that for a given λ_j , $\psi(\lambda_j)$ decomposes into K separable transportation problems (see Chapter II). Concavity of $\psi(\lambda_j)$ is preserved due to the fact that the sum of concave functions is concave. Hillier and Lieberman (17) have proved the concavity of $\psi(\lambda_j)$).

Let (\bar{z}, \bar{u}) be the pseudo-global point we are working with and let $\phi(\bar{z}, \bar{u}) = \bar{k} \geq k$. An intuitive explanation for the approach and its efficiency is given here. As λ_j increases from zero, $\psi(\lambda_j)$ increases initially and then begins to decrease as the term involving λ_j dominates the constant term d . As a result, there are very few breakpoints after $\psi(\lambda_j) < k$ and this makes our approach very attractive.

For $\lambda = \lambda_j$, let the solution of the K transportation problems as in the last section give $\psi(\lambda_j)$ at $u_j \in U$. Then

$$\begin{aligned}\psi(\lambda_j) &= d^t u_j + (\bar{z} - \bar{e}^j \lambda_j)^t D u_j. \\ \therefore \frac{\partial \psi(\lambda_j)}{\partial \lambda_j} &= -(\bar{e}^j)^t D u_j = m_j, \text{ say.}\end{aligned}$$

$\psi(\lambda_j)$ is easily seen to be piecewise linear with breakpoints occurring whenever a change in allocations occurs. Hence, at the breakpoint, we let the above derivative represent the slope of $\psi(\lambda_j)$ on the portion corresponding to allocation u_j .

$$\text{But } Du_j = \begin{bmatrix} A \\ A \\ A \\ A \\ 0_{2n \times 1} \end{bmatrix} \quad \text{where } A = \begin{bmatrix} \sum_{k=1}^K u_{1,1,k} \cdot t_k \\ \vdots \\ \sum_{k=1}^K u_{1,n+m,k} \cdot t_k \\ \sum_{k=1}^K u_{2,1,k} \cdot t_k \\ \vdots \\ \sum_{k=1}^K u_{2,m+n,k} \cdot t_k \\ \vdots \\ \sum_{k=1}^K u_{n,n+m,k} \cdot t_k \end{bmatrix}_{n(n+m) \times 1}.$$

Hence, if we are working for $\bar{\lambda}$ for x_{za}^+ ,

$$m_j \text{ for } x_{za}^+ = \sum_{k=1}^K \cdot \sum_{\substack{j \in \{1, \dots, n+m\} \\ x_j \leq x_z}} (u_{zjk} + u_{jzk}) \cdot t_k \\ - \sum_{k=1}^K \cdot \sum_{\substack{j \in \{1, \dots, n+m\} \\ x_j > x_z}} (u_{zjk} + u_{jzk}) \cdot t_k$$

$$\text{and } m_j \text{ for } x_{za}^- = \sum_{k=1}^K \cdot \sum_{\substack{j \in \{1, \dots, n+m\} \\ x_j \geq x_z}} (u_{zjk} + u_{jzk}) \cdot t_k \\ - \sum_{k=1}^K \cdot \sum_{\substack{j \in \{1, \dots, n+m\} \\ x_j < x_z}} (u_{zjk} + u_{jzk}) \cdot t_k$$

4.6.1 Algorithm to Solve Problem PAR 1

Step 0: Consider determining $\bar{\lambda}_j$ for x_{za}^+ (x_{za}^-). If $x_a \geq x_j$ ($x_a \leq x_j$)
 $\forall j \in \{n+1, \dots, n+m\}$, x_{za}^+ (x_{za}^-) will always be zero. Hence we can
 arbitrarily set $\bar{\lambda}_j = \infty$.

Step 1: Let $\lambda_1 = L$, a large number and let k be the current best value
 for $\phi(z, u)$.

Step 2: Determine $\theta_1 = \psi(\lambda_1)$ {As in the last section}.

Step 3: If $\theta_1 \geq k$, STOP with $\bar{\lambda}_j = \infty$ ($\equiv L$). Else, go to step 4.

Step 4: Determine m_j from the expression in the last section.

Step 5: Determine $\theta_2 = \psi(\lambda_2)$ where $\lambda_2 = \frac{k - \theta_1 + m_j \lambda_1}{m_j}$

If $\theta_2 = k$, STOP; $\bar{\lambda}_j = \lambda_2$

If $\theta_2 < k$, let $\lambda_1 = \lambda_2$, $\theta_1 = \psi(\lambda_2)$ and go to step 4.

4.6.2 Modification for Problem PAR 2

Here, use $\theta = \bar{\psi}(\lambda_j)$ instead of $\psi(\lambda_j)$ as above, and for calculating
 the slopes, the negative of the expressions of Section 4.6 must be
 used.

4.6.3 Convergence of the Algorithm of Section 4.6.1

It may be noted that the slopes are finite and negative and
 that every time we go through step 4, we are on a different linear
 portion of $\psi(\lambda_j)$ with a strictly decreasing absolute value of the slope.
 Also, since the extreme points of U are finite, so are the number of
 breakpoints. Hence convergence is guaranteed.

4.7 Determination of a Weak Pseudo Global Minimum

We have shown that a valid cutting plane can be developed from a

pseudo global minimum. It may seem that we would now be required to solve the location problem with the cuts as additional constraints. Fortunately, the special structure of the set Z allows us to relax the necessity of a pseudo global minimum to what has been called a weak pseudo global minimum (37). Note that decomposition techniques which involve convex combinations of different points are likely to violate the constraints of the type $x_{ij}^+ \cdot x_{ij}^- = 0$. Restricted basis entry would, if imposed here, further complicate matters.

Let $A(\bar{z})$ denote the set of adjacent extreme points of \bar{z} . Also at stage s , let $g^s(z) \geq 1$ denote the s cuts $g_i(z) \geq 1$, $i = 1, \dots, s$ we have introduced into the set Z .

4.7.1 Definition

At stage s , let \bar{z} be an extreme point of Z and \bar{u} of U such that $g^s(\bar{z}) \geq 1$, and $\min_{u \in U} \phi(\bar{z}, u) = \phi(\bar{z}, \bar{u})$. Then (\bar{z}, \bar{u}) is said to be a weak pseudo global minimum of $\phi(z, u)$ with respect to the cuts $g^s(z) \geq 1$ if for each $\hat{z} \in A(\bar{z})$ such that $g^s(\hat{z}) \geq 1$, $\min_{u \in U} \phi(\hat{z}, u) \geq \phi(\bar{z}, \bar{u})$.

4.7.2 Lemma

Let $g_i(z) \geq 1$, $i = 1, \dots, s$ be the s cutting planes generated thus far. Let k be the objective function value for the current best solution. If $g_p(\hat{z}) < 1$ for some $\hat{z} \in A(\bar{z})$ and some $p \in \{1, \dots, s\}$, then $\min_{u \in U} \phi(\hat{z}, u) \geq k$.

Proof: Let \bar{k} be the value of the best solution when the cut $g_p(z) \geq 1$ was generated. Then $\bar{k} \geq k$. Also, $g_p(\hat{z}) < 1$ implies that $\min_{u \in U} \phi(\hat{z}, u) \geq \bar{k}$ from the definition of a valid cut. Since $\bar{k} \geq k$, the result follows.

4.7.3 Theorem

A valid cutting plane can be generated from a weak pseudo global minimum (\bar{z}, \bar{u}) .

Proof: In Theorem 4.3.3, we showed how a valid cutting plane could be generated from a pseudoglobal minimum. The only characteristic of the pseudo global point which we used was that it implied that $\bar{\lambda}_j > 0$, $j \in J$, $J_1 \neq \{\phi\}$. Hence it is sufficient to show that this property is also true for a weak pseudo global minimum.

Consider first, $j \in J_1$. Let $\hat{\lambda}_j > 0$ be such that $\hat{z} = \bar{z} - e^j \hat{\lambda}_j \in A(\bar{z})$. Then if $g_i(\hat{z}) \geq 1$ for $i=1, \dots, s$ (where s cuts have been generated thus far), the argument in Lemma 4.3.2 yields $\bar{\lambda}_j > 0$. Now suppose that $g_p(\hat{z}) < 1$ for some $\hat{z} \in A(\bar{z})$ and some $p \in \{1, \dots, s\}$. As before, let $\psi(\lambda_j) = \min_{u \in U} \phi(\bar{z} - e^j \lambda_j, u)$ for $\lambda_j \geq 0$. Hence, if $\hat{z} = \bar{z} - e^j \hat{\lambda}_j$, $\hat{\lambda}_j > 0$, then from Lemma 4.7.2, $\psi(\hat{\lambda}_j) \geq k$. Hence $\bar{\lambda}_j = \max_{\lambda_j} \{\psi(\lambda_j) \geq k\} \geq \hat{\lambda}_j > 0$. Hence for $j \in J_1$, $\bar{\lambda}_j > 0$.

Also, for $j \in J_2$ if $J_1 \neq \{\phi\}$, the argument in Lemma 4.3.2 leads to $\bar{\lambda}_j > 0$. Hence, a valid cutting plane may be generated from a weak pseudo global minimum.

4.7.4 Algorithm to Find a Weak Pseudo Global Minimum

We will develop this algorithm based on the definition 4.7.1.

Step 1: At stage s , find an extreme point $z_s \in Z$ feasible to the cuts $g^s(z) \geq 1$. If none exists, STOP - the current best solution is optimum. Otherwise GO TO STEP 2. (Section 4.9 shows how to determine such an extreme point.)

Step 2: Find a $\bar{z} \in A(z_s)$ such that

$$g^s(z) \geq 1 \quad \text{and}$$

$$\min_{u \in U} \phi(\bar{z}, u) < \min_{u \in U} \phi(z_s, u).$$

If no such point exists, solve

$$\min_{u \in U} \phi(z_s, u) = \phi(z_s, \bar{u}) \text{ and terminate with } (z_s, \bar{u}) \text{ as the weak}$$

pseudo global minimum.

If such a point exists, GO TO STEP 3.

Step 3: Replace z_s by \bar{z} and GO TO STEP 2.

Since (i) The cardinality of $A(\bar{z})$, $\bar{z} \in Z$ is finite; (ii) The objective function value of $\phi(z, u)$ is bounded below and (iii) Each sequence of steps 2 and 3 results in a strict decrease of $\phi(z, u)$, the algorithm is finitely convergent.

4.8 Determination of a Good Starting Solution

In problems of this type where a local optimum is not necessarily a global optimum, the need for a good starting solution can never be overemphasized. Considerable reduction in computational times may be obtained and some advantageous simplifications can also result (see Section 6.2.6). Hence, any extra effort spent here is worthwhile.

For the case of a single product with no interactions between sources, reference [37] gives a good starting solution. However, with multiple products and interactions between sources, it becomes difficult to exploit all possibilities. A reasonably good procedure is given here:

Step 1: Initialize by putting $p = 1$.

Step 2: At stage p ($1 \leq p \leq n$), consider the positioning of the p^{th} new facility. Let

$$a_{pk'} = \max_{k \in \{1, \dots, k\}} a_{pk}$$

and let

$$b_{qk'} = \max_{i \in \{n+1, \dots, n+m\}} b_{ik'}$$

Put $(x_p, y_p) = (x_q, y_q)$

Replace b_{qk} by $b_{qk} + b_{pk}$ for $k = 1, \dots, K$,
 $k \neq k'$

Replace $b_{qk'}$ by $b_{qk'} - a_{pk'} + b_{pk'}$

GO TO STEP 3.

Step 3: Replace b by $b+1$. If b is greater than n ,

STOP - we have a starting solution; else, GO TO STEP 2.

Note: The adjustments in the demands (b_{ik}) and the capacities (a_{ik}) made above are ONLY for the purpose of the starting solution. Hence, if any b_{ik} becomes negative above, it is inconsequential.

4.9 Determination of a Feasible Extreme Point to the Set of Cuts

The main difficulty here is to guarantee that the constraints $x_{ij}^+ \cdot x_{ij}^- = 0$ hold implicitly. It is because of this that the following standard procedures are invalid: (i) Solving the location problem with the cuts as additional constraints (ii) Using decomposition techniques with the cuts forming a set \bar{Z} and the constraint $z \in \bar{Z}$. (iii) Use of separable programming. An efficient method which guarantees the determination of a feasible extreme point will be developed here.

When solving the location problems P_x and P_y , we had discussed the case where some existing facilities had the same x or y coordinates. Let the modifications suggested there (Sec. 3.2.2) result in m_x and m_y equivalent existing facilities in the x and y coordinate sets respectively. (i.e. any new facility can have one of the m_x 'x coordinate locations' and

one of the myth y coordinate locationsth).

For $i = 1, \dots, n$ let

$$\begin{aligned} x_i &= \sum_{j=n+1}^{n+mx} \lambda_{ij1} x_j & y_i &= \sum_{j=n+1}^{n+my} \lambda_{ij2} y_j \\ &\text{and} & & \\ \sum_{j=n+1}^{n+mx} \lambda_{ij1} &= 1 & \sum_{j=n+1}^{n+my} \lambda_{ij2} &= 1 \\ \lambda_{ij1} &= 0 \text{ or } 1 \quad \forall i, j & \lambda_{ij2} &= 0 \text{ or } 1 \quad \forall i, j. \end{aligned}$$

Let the s cuts generated be $g^p(z) \geq 1 \quad p = 1, \dots, s$. Then, finding a feasible extreme point to these cuts reduces to finding a set of λ_{ijk} ($i = 1, \dots, n$, $j = n+1, \dots, mx+n$ if $k=1$, $j = n+1, \dots, n+my$ if $(k=2)$) such that

$$\sum_{i=1}^n \left[\sum_{j=n+1}^{n+mx} g_{ij1}^p \lambda_{ij1} + \sum_{j=n+1}^{n+my} g_{ij2}^p \lambda_{ij2} \right] \geq 1 \quad (p=1, \dots, s)$$

$$\sum_{j=n+1}^{n+mx} \lambda_{ij1} = 1$$

$$\sum_{j=n+1}^{n+my} \lambda_{ij2} = 1 \quad i = 1, \dots, n$$

$$\lambda_{ijk} = 0 \text{ or } 1 \quad \forall i, j, k.$$

Here, $g_{ij1}^p = g^p(z) \big|_{x_i=x_j, x_{\bar{p}t}^{\pm} = 0 \text{ for } \bar{p} \neq i, \forall t \text{ and } y_{qr}^{\pm} = 0 \forall q, r}$.

and, $g_{ij2}^p = g^p(z) \big|_{y_i=y_j, x_{\bar{p}t}^{\pm} = 0 \forall \bar{p}, t, y_{qr}^{\pm} = 0 \text{ for } q \neq i, \forall r}$.

4.9.1 Definition

For each combination of i and k , we define a block to be a set

containing the terms $g_{ijk}^p \cdot \lambda_{ijk}$ (for $j = n+1, \dots, n+mx$ if $k=1$ and $j = n+1, \dots, n+my$ if $k=2$). Hence, there are $2n$ blocks and each block has precisely one $\lambda_{ijk} = 1$ and all other $\lambda_{ijk} = 0$.

$$\text{Let } \bar{g}^t = \max_{z \in Z} g^t(z) \quad \text{for } t = 1, \dots, s.$$

\bar{g}^t is obtained trivially by letting that $\lambda_{ijk} = 1$ in each block for which g_{ijk}^t is maximum in that block, and putting all other $\lambda_{ijk} = 0$.

$$\text{Then, defining } g_{ijk} = \sum_{p=1}^s \frac{1}{\bar{g}^p} g_{ijk}^p,$$

$$\sum_{t=1}^s \frac{g^t(z)}{\bar{g}^t} = \sum_{i=1}^n \left[\sum_{j=n+1}^{n+mx} g_{ij1} \cdot \lambda_{ij1} + \sum_{j=n+1}^{n+my} g_{ij2} \cdot \lambda_{ij2} \right].$$

We now form $2n$ blocks with the terms $g_{ijk} \cdot \lambda_{ijk}$ as before. In each block, we rearrange the terms in ascending order of magnitude of g_{ijk} .

Let the i^{th} term in the j^{th} block be renamed as $g_{ij} \gamma_{ij}$ where g_{ij} is associated with the corresponding g_{pqk} and γ_{ij} with λ_{pqk} for some p, q, k (Here, $i = 1, \dots, mx$ if $k=1$ and $i = 1, \dots, my$ if $k=2$ and $j = 1, \dots, 2n$.) For the j^{th} block, let $k_j = mx$ or my according as $k=1$ or 2 for that block. Also, for $k=1$, let $j = 1, \dots, n$ and for $k=2$, let $j = n+1, \dots, 2n$. We now have,

$$\sum_{t=1}^s \frac{g^t(z)}{\bar{g}^t} = \sum_{j=1}^{2n} \sum_{i=1}^{k_j} g_{ij} \gamma_{ij}$$

where, $g_{pj} \geq g_{qj}$ for $k_j \geq p \geq q \geq 1$

and $\sum_{i=1}^{k_j} \gamma_{ij} = 1$ for $j = 1, \dots, 2n$

$$\gamma_{ij} = 0 \text{ or } 1 \quad \text{for } \forall i, j.$$

We now formulate PROBLEM P_E :

$$\begin{aligned} \underline{\text{max.}} \quad & \sum_{j=1}^{2n} \sum_{i=1}^{k_j} g_{ij} \gamma_{ij} \\ \underline{\text{subj. to}} \quad & \sum_{i=1}^{k_j} \gamma_{ij} = 1 \quad (j=1, \dots, 2n) \\ & \gamma_{ij} \geq 0 \quad \forall i, j. \end{aligned}$$

The following are evident for this problem:

(i) Problem P_E is a linear programming problem. The number of basic variables here equals the number of constraints (excluding non-negativity constraints which is equal to $2n$). Here, only one $\gamma_{ij} > 0$ in each constraints and this γ_{ij} must be equal to 1. This implies that every basic feasible solution to P_E corresponds to an extreme point $z \in Z$. In fact, for $j=1, \dots, n$

$$\gamma_{pj} = 1 \text{ implies } x_j = d_p \text{ and for } j = n+1, \dots, 2n,$$

$\gamma_{pj} = 1$ implies $y_{j-n} = e_p$. We will denote a basic feasible solution to problem P_E by a $2n$ dimensional vector $(\delta_1, \dots, \delta_t, \dots, \delta_{2n})$

where

$$\gamma_{\delta_j, j} = 1 \text{ for } j = 1, \dots, 2n \text{ is the particular solution.}$$

(ii) As before, for each $j = 1, \dots, 2n$

$$g_{pj} \geq g_{qj} \quad \text{for } k_j \geq p \geq q \geq 1.$$

(iii) Since the objective function value is $\sum_{t=1}^s \frac{g^t(z)}{g^t}$, this must be

$$\geq \sum_{t=1}^s 1/g^t \quad \text{if the corresponding solution } z \text{ is to satisfy } g^t(z) \geq 1$$

$t = 1, \dots, s$. (This is a necessary though not sufficient condition.)

We will now describe a procedure to rank the extreme points of problem P_E till a solution \hat{z} is obtained for which all the cuts are

satisfied or till $\sum_{t=1}^s \frac{g^t(\hat{z})}{\bar{g}^t} < \sum_{t=1}^s 1/\bar{g}^t$ which would indicate that no

extreme point is feasible to the set of cuts. An intuitive explanation for the choice of this objective function for problem P_E will now be given. If instead, we had decided to use the same strategy with the objective function $\max \sum_{t=1}^s g^t(z)$, it is quite likely that a subset of the s cuts could have been oversatisfied by some solutions and we would have to rank several points before stopping. In our approach, by dividing each cut by its maximum we will more readily locate a solution almost equidistant from each cut. In what follows, we will make use of the following expressions and their accompanying connotations:

- (i) "picked-up" - Selected from the list of extreme points as the next best solution.
- (ii) "listing of adjacent points" - Let the solution we pick-up be $(\delta_1, \dots, \delta_{2n})$. Then, for $i=1, \dots, 2n$ we will generate and store the adjacent extreme points $(\delta_1, \dots, \delta_{i-1}, \dots, \delta_{2n})$ (If $\delta_i = 1$, no adjacent point corresponding to it is listed.) Let the set of adjacent extreme points to the k^{th} solution we pick-up be A^k . (Note that, we list at most $2n$ of the possible $n(mx-1) + n(my-1)$ adjacent extreme points.)
- (iii) "Solution is better off (worse off)" - The objective function value for problem P_E is \geq (\leq) that value we are comparing it with.

4.9.2 Lemma

If an l.p. problem has a $(k+1)$ st best extreme point solution, then it is adjacent to some element of the set containing the {optimal, second

best, ..., k^{th} best} extreme point solutions of the problem.

Proof: See (20).

4.9.3 Lemma. At the k^{th} stage, suppose that we have,

(i) The optimal, second best, ..., k^{th} best extreme point solutions N^1, N^2, \dots, N^k .

(ii) A list L_k of adjacent extreme points where,

$$L_k = \bigcup_{q=1}^k A^q$$

Then, the $(k+1)^{\text{st}}$ best extreme point solution $N^{k+1} \in L_k$.

Proof: We will prove this by contradiction. Let, if possible,

$N_{k+1} \notin L_k$. Let $N^{k+1} = (\xi_1, \dots, \xi_t, \dots, \xi_{2n})$. Then, by Lemma 4.9.3, N^{k+1} is adjacent to at least one of the points $N^z, z \in [1, \dots, k]$. Let N^{k+1} be adjacent to N^p and let $N^p = (\alpha_1, \dots, \alpha_t, \dots, \alpha_{2n})$.

By property (i) (Section 4.9.1) of the l.p. problem P_E , N^p and N^{k+1} differ in only one, say the t^{th} , component. Also, $\alpha_t > \xi_t$, for if not, then $g_{\xi_t, t} \geq g_{\alpha_t, t}$ by the ordering property and $g_{\xi_j, j} = g_{\alpha_j, j}$ for $j \in [1, \dots, 2n], j \neq t$. This implies that the objective function value for $N^{k+1} = \sum_{j=1}^{2n} g_{\xi_j, j} \geq \sum_{j=1}^{2n} g_{\alpha_j, j}$ = that for N^p which is a contradiction.

Suppose then that $\xi_t = \alpha_t - 1$

$$\xi_j = \alpha_j \text{ (for } j \in \{1, \dots, 2n\}, j \neq t)$$

But by definition, this means that $N^{k+1} \in A^p$ and since $A^p \subseteq L_k$, $N^{k+1} \in L_k$ which is a contradiction.

Hence, $\xi_t = \alpha_t - r$ where $2 \leq r \leq (\alpha_t - 1)$. Consider the point $N^{q1} = (\alpha_1, \dots, \alpha_t - 1, \dots, \alpha_{2n})$.

Case (i): N^{q_1} has not been picked up. Then, since

$$\alpha_t - 1 > \xi_t, \quad \therefore g_{\alpha_t-1,t} \geq g_{\xi_t,t}$$

and hence as before, N^{q_1} is better off than N^{k+1} , which is a contradiction since we have assumed that N^{k+1} is the $(k+1)$ st best. Hence, Case (ii) must be true.

Case (ii): N^{q_1} has been picked up. This means that N^{q_2} is listed, where

$$N^{q_2} = (\alpha_1, \dots, \alpha_t-2, \dots, \alpha_{2n}) \quad (\because N^{q_2} \in A^{q_1}).$$

Since $N^{k+1} \notin L_k$, $N^{k+1} \neq N^{q_1}$ or N^{q_2} .

$$\therefore \xi_t = \alpha_t - r \text{ where } 3 \leq r \leq \alpha_t - 1.$$

Replacing N^{q_1} by N^{q_2} and repeating the above argument $\alpha_t - 3$ more times, we arrive at the conclusion that either (a) $N^{k+1} \in L_k$ or (b) $\xi_t = \alpha_t - r$ such that $\alpha_t \leq r \leq \alpha_t - 1$. Clearly, (b) is impossible and hence, $N^{k+1} \in L_k$.

4.10 Algorithm "Feas"

Suppose we are to find a feasible extreme point z of the set Z which satisfies the s cuts $g^s(z) \geq 1$. Let the objective function value of problem P_E be denoted by ϕ_{P_E} . Calculate as before

$$\bar{g} = \sum_{t=1}^s 1/\bar{g}^t.$$

Step 1: Determine ϕ_{P_E} for the optimum solution $N^1 = (k_1, \dots, k_t, \dots, k_{2n})$ to problem P_E . If $\phi_{P_E} < \bar{g} \Rightarrow \text{STOP}$: there is no feasible point to the system of cuts. Else, go to step 2 with this solution as the first solution picked up.

Step 2: At the k^{th} stage, suppose we have $L_{k-1} = \bigcup_{q=1}^{k-1} A^q$. Pick up the k^{th} best solution $N^k \in L_{k-1}$. (Drop this and pick another if this has already been picked up in case of alternate k^{th} optimal solutions).
GO TO STEP 3.

Step 3: For z_k corresponding to N^k , calculate for $t = 1, \dots, s$

(a) If $\sum_{t=1}^s \frac{g^t(z_k)}{\bar{g}^t} < \bar{g}$, STOP - no extreme point of Z is feasible to the system of cuts.

(b) If $g^t(z_k) \geq 1 \ \forall t$, STOP - $z_k \in Z$ is a feasible extreme point.

(c) If (a) and (b) do not hold, generate A^k and let $L_k = L_{k-1} \cup A^k$.

Replace k by $k+1$ and GO TO STEP 2.

4.10.1 Guarantee of Finite Convergence to a Feasible Point if it Exists

Since the number of extreme points we list are finite and no extreme point is repeated the procedure is finitely convergent.

Since we stop either when we find a feasible extreme point or when we have listed all points with $\phi_P \geq \bar{g}$, finding a feasible point is guaranteed. This is so because there is no solution \hat{z} with $g^t(\hat{z}) \geq 1$

for $t = 1, \dots, s$ and $\sum_{t=1}^s \frac{g^t(\hat{z})}{\bar{g}^t} > \sum_{t=1}^s \frac{1}{\bar{g}^t} = \bar{g}$.

We now have all the tools we require to solve our location-allocation problem. A statement of the complete algorithm and some illustrative examples will be presented next.

CHAPTER V

THE ALGORITHM TO SOLVE A GENERAL RECTILINEAR-DISTANCE LOCATION-ALLOCATION PROBLEM

In this chapter, we will present the complete algorithm to solve problem GRLAP and will illustrate the procedure through some examples.

5.1 Statement of the Algorithm

Step 1: Using the approach of section 4.8, locate the sources to give a starting solution.

Step 2: Using this location, solve the pure allocation problem of Section 2.5.1.

Step 3: With this allocation, solve the pure location problem of Section 2.5.2 using Algorithm MFL0C of section 3.5.

Step 4: If this location is the same as that used in step 2, GO TO STEP 5. Else go to STEP 2.

Step 5: Use the algorithm of Section 4.7.4 on this local star minimum. The resulting solution is hence a pseudo global minimum. Store its location, allocation and objective function value as the current best solution.

Step 6: Generate the first cutting plane based on this pseudo global minimum. If $J_1 = \{\phi\}$, STOP - the current best solution is optimum.

Step 7: Use algorithm FEAS of Section 4.10 to determine an extreme point feasible to the system of cuts. If none exists, STOP - the current best solution is OPTIMAL. Otherwise, GO TO STEP 8.

Step 8: Starting from this point, determine a weak pseudo global minimum according to the method of Section 4.7.4.

Step 9: Determine the objective function value for this point. If it is better than the current best, store its location, allocation and objective function value as the current best solution. GO TO STEP 10.

If it is not better than the current best point, GO TO STEP 10.

Step 10: Generate another cut based on this weak pseudo global point. GO TO STEP 7. (If $J_1 = \{\phi\}$, STOP - the current best solution is optimum.)

5.2 Illustrative Examples

5.2.1 Illustrative Example 1

Using the notations introduced in Chapter 1, let

$$n = 2, m = 4, k = 2, \quad a_{ik} = \begin{bmatrix} 25 & 25 \\ 50 & 0 \end{bmatrix}, \quad b_{ik} = \begin{bmatrix} 0 & 0 \\ 0 & 5 \\ 15 & 5 \\ 30 & 5 \\ 20 & 5 \\ 10 & 5 \end{bmatrix}, \quad c_{ik} = \begin{bmatrix} 3 & 1 \\ 2 & 0 \end{bmatrix};$$

Let us number the new facilities as 1 and 2 and let the existing facilities be numbered 3,4,5,6 with their position being

$$\begin{array}{cc} x_i & y_i \\ 3 & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ 4 & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ 5 & \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ 6 & \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{array} \quad . \quad \text{Let } t_k = [1,2] \quad .$$

Solution:

Step 1: According to Section 4.8, the starting solution is easily seen to be $(x_1, y_1) = (0, 1)$ and $(x_2, y_2) = (1, 1)$. {Henceforth, we will write this as: 1(0,1) and 2(1,1)}.

Step 2: For this location, we solve the allocation problems for the two products using the cost coefficients as $c_{ik} + t_k d(i,j)$:

Prod. #1:

	1	2	3	4	5	6	Supply
1	3 0	4 	4 15	3 10	4 	5 	25
2	3 	2 0	4 	3 20	2 20	3 10	50
demands	0	0	15	30	20	10	

Prod. #2

	1	2	3	4	5	6	Supply
1	1 0	3 5	3 5	1 5	3 5	5 5	25
2	2 	0 	4 	2 	0 0	2 	0
demands	0	5	5	5	5	5	

Sum of Obj. Fn. Values for the 2 products = 295 $\equiv \phi$.

Step 3: The location problem MFL0C has now to be solved based on these allocations. The pertinent coefficients are $w_{ij}^1 = \sum_{k=1}^2 t_k \cdot u_{ijk}$.

$$\therefore w_{ij}^1 = \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ \begin{matrix} 1 \\ 2 \end{matrix} & \begin{bmatrix} 0 & 10 & 25 & 20 & 10 & 10 \\ 0 & 0 & 0 & 20 & 20 & 10 \end{bmatrix} \end{matrix}$$

Prob. P_x:

Here, we let a represent facilities 3 and 4 and b \equiv 5 and 6.

As before,

$$w_{ij} = \begin{array}{c} \begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix} \end{array} \begin{array}{cc} a & b \\ 45 & 20 \\ 20 & 30 \end{array} \end{array}$$

The current location is $x_1 = x_a$ and $x_2 = x_b$.

$$\therefore r_1 = -15, \ell_1 = -75, r_2 = -50, \ell_2 = -10.$$

$$\because r_i \leq 0, \ell_i \leq 0 \ (i=1,2) \text{ and } x_1 \neq x_2$$

this solution is optimum.

Prob. P_y: Here, we let a represent the combined facilities 3 and 6 and

let b represent 4, and 5.

$$\therefore w_{ij} = \begin{array}{c} \begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix} \end{array} \begin{array}{cc} a & b \\ 35 & 30 \\ 10 & 40 \end{array} \end{array}$$

Here, $y_1 = y_2 = y_b$.

Also, $r_1 = -75, \ell_1 = -5, r_2 = -50, \ell_2 = -30$. Hence, no single movements are advantageous

For joint movements,

$$S_h = \{1,2\} \text{ and } A = \begin{array}{c} \begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} 0 & 10 \\ 10 & 0 \end{bmatrix} \end{array}$$

Hence, we have, new facility i:

$$\begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} \ell_i \\ u_i \\ c_i \end{array} & \begin{array}{cc} -5 & -30 \\ 10 & 10 \\ = \ell_i + u_i & = 5 \quad -20 \end{array} \end{array}$$

<u>Ranked Extreme Points:</u>	$g(x)$	$f(x)$	Bounds on f^* :
(1,0)	5	-5	[5,-5]
(0,0)	0	0	[0,0]

Hence, since $f^* = 0$, no joint movements are possible. Hence we are optimum with respect to both the x and y coordinate locations.

Step 4: The current location is 1(0,1) and 2(1,1) which is the same as in step 2. This is hence a local star minimum.

Step 5: We now obtain a pseudo global minimum by using the algorithm of Section 4.7.4 with $S = 0$. For this purpose, we have to evaluate (by solving the allocation problems as in Step 2) all positions obtained by moving each new facility to all adjacent locations possible, holding the other facilities fixed. The current best (for the local star minimum) is from Step 2, $\phi = 295$

i	(x_1, y_1)	(x_2, y_2)	ϕ_i	
1)	(1,1)	(1,1)	310	
2)	(0,0)	(1,1)	300	$\because \phi_i \geq 295 \forall i$, the local star minimum is also a pseudo global minimum.
3)	(0,1)	(0,1)	295	
4)	(0,1)	(1,0)	305	

Step 6: We now generate a cut based on $k = 295$ and the point 1(0,1) and 2(1,1). According to Section 4.5, we now have to determine $\bar{\lambda}_j$ $j \in J_1 \cup J_2$ for x_{14}^\pm , y_{14}^\pm , x_{25}^\pm and y_{25}^\pm . Consider determining $\bar{\lambda}_j$ for x_{14}^+ :

Section 4.5 gives us the following two transportation problems to evaluate $\psi(\lambda_j)$ for problem PAR 1:

Product #1

	1	2	3	4	5	6	Supplies
1	3	$4-\lambda$	$4+\lambda$	$3+\lambda$	$4-\lambda$	$5-\lambda$	25
2	$3-\lambda$	2	4	3	2	3	50
demands	0	0	15	30	20	10	

Product #2

	1	2	3	4	5	6	Supplies
1	1	$3-2\lambda$	$3+2\lambda$	$1+2\lambda$	$3-2\lambda$	$5-2\lambda$	25
2	$2-2\lambda$	0	4	2	0	2	0
demands	0	5	5	5	5	5	

For $\lambda =$ a large number L , the above solution is obtained with the total objective function value $\theta = 345 - 35L$. Hence with $L \rightarrow \infty$, $\theta \rightarrow -\infty < 295$. This means $\bar{\lambda}_j$ is finite. Calculating m_j as in section 4.6.1, the slope is -35 (this is simply the sum of the products of the allocations and the coefficients of λ). Hence, $\lambda_2 = \frac{295 - (345 - 35L) - 35L}{-35}$ or $\lambda_2 = 1.4285714$. Putting $\lambda = \lambda_2$ and resolving the above transportation problems, we obtain $\theta = 295$. $\therefore 1/\bar{\lambda}_j = 1/1.4285714 = 0.7$.

Also, by Step 0 of algorithm in Section 4.6.1, since $x_4 \leq x_i$ for $i = 3, 4, 5, 6$, $\bar{\lambda}_j$ for x_{14} is arbitrarily put at ∞ .

Similarly, we have

Non-basic Variable	$j \in J_1$ $\bar{\lambda}_j$	$j \in J_2$ $\bar{\lambda}_j$
x_{25}^+	∞	∞
x_{25}^-	1	-
y_{14}^+	∞	∞
y_{14}^-	0.75	-
y_{25}^+	∞	∞

Consider, determining $\bar{\lambda}_j$ for y_{25}^- .

Formulating the transportation problems as above, it can be seen that as $\lambda \rightarrow \infty$, $\psi(\lambda) \rightarrow \infty > 295$. Hence $\bar{\lambda}_j = \infty$. We hence have to determine $\bar{\lambda}_j$, $j \in J_2$ for this from Problem PAR 2. The transportation problems to evaluate $-\bar{\psi}(\lambda_j)$ are:

Prod. #1:

	1	2	3	4	5	6	Supplies
1	-3 0	-4+ λ	-4	-3 5	-4 20	-5	25
2	-3+ λ	-2 0	-4- λ 15	-3+ λ 25	-2+ λ	-3- λ 10	50
demands	0	0	15	30	20	10	

Product #2:

	1	2	3	4	5	6	Supplies
1	-1 0	-3+2 λ 5	-3 5	-1 5	-3 5	-5 5	25
2	-2+2 λ	0	-4-2 λ 0	-3+2 λ	-2+2 λ	-3-2 λ	0
demands	0	5	5	5	5	5	

When $\lambda_j =$ a large number L , the above solution is obtained for which $-\bar{\psi}(\lambda_j = L) = -335 + 10L$. Hence, as $\lambda_j = L \rightarrow \infty$, $\bar{\psi}(\lambda_j) \rightarrow \infty$ and hence, $\bar{\lambda}_j$ is finite. According to Sections 4.6.1 and 4.6.2, the slope here is $= -10$ and $\lambda_2 = (295 - 335 + 10L - 10L)/-10 = 4$. Putting $\lambda = 4$ and re-solving the above two transportation problems, we obtain $\bar{\psi}(4) = 295$. Hence, $\bar{\lambda}_j = 4$ for y_{25}^- ($j = J_2$). The first cut is:

$$0.7 x_{14}^+ + x_{25}^- + 0.75 y_{14}^- - 0.25 y_{25}^- \geq 1.$$

Step 7: A feasible point to this by ALGORITHM FEAS is $1(1,0)$ and $2(0,1)$.

(A detailed use of this algorithm is given in the next example.)

Step 8: Working as in Step 5, we obtain a weak pseudo global minimum: $1(0,0)$ and $2(0,1)$.

Step 9: The value of ϕ for this is 280. This is hence the current best solution.

Step 10: Another cut may now be generated based on this point as:

$$0.5 x_{13}^+ + 0.7 y_{13}^+ + 0.2 y_{24}^- \geq 1.$$

A feasible point to these 2 cuts is: $1(1,1)$ and $2(0,0)$. The weak pseudo global point (WPG) is $1(1,1)$ and $2(0,1)$ with $\phi = 280$. This is hence, alternate optimal relative to the current best solution. A third cut is then given by:

$$0.7 x_{15}^- + 0.4 x_{24}^+ + 0.5 y_{15}^- - 0.2 y_{24}^- \geq 1.$$

It can be shown that no extreme point is feasible to these 3 cuts, and we stop with the optimal location $\equiv 1(0,0), 2(0,1)$; the optimum

$$\text{allocations of product \#1: } \begin{array}{c} 1 \\ 2 \end{array} \begin{array}{c} 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \\ \left[\begin{array}{cccccc} 0 & 0 & 15 & 0 & 0 & 10 \\ 0 & 0 & 0 & 30 & 20 & 0 \end{array} \right] \end{array}$$

$$\text{and of product \#2 as } \begin{array}{c} 1 \\ 1 \end{array} \begin{array}{c} 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \\ \left[\begin{array}{cccccc} 0 & 5 & 5 & 5 & 5 & 5 \end{array} \right] \end{array}$$

with optimum $\phi = 280$.

5.2.2 Illustrative Example 2

We consider here the location of two facilities (1,2) with capacities 40 and 30 units of a single product and determine the allocations to 5 existing facilities (3,4,5,6,7) located at (0,0), (0,2), (2,2), (2,0) and (1,1) with requirements of 10, 20, 5, 15, and 10 units respectively.

Solution: The general technique is the same as above and we solve this to illustrate mainly, the application of ALGORITHM FEAS.

$$\text{Here, } n = 2, m = 5, k = 1, t_1 = 1, c_1 = 0, a_{ik} = \begin{bmatrix} 40 \\ 30 \end{bmatrix}, b_{ik} = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 20 \\ 5 \\ 15 \\ 10 \end{bmatrix}$$

$$\text{and locations of existing facilities: } \begin{array}{c} 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} x_i, y_i \\ \left[\begin{array}{cc} 0, & 0 \\ 0, & 2 \\ 2, & 2 \\ 2, & 0 \\ 1, & 1 \end{array} \right] \end{array}$$

Starting Solution: 1(0,2) and 2(2,0).

Local Star minimum: 1(0,2) and 2(1,0).

Pseudo global minimum: $1(0,2)$ and $2(1,0)$ with $\phi = 50$.

The first cut based on this is (with $k = 50$).

$$0.272727 x_{14}^+ + 1 \cdot x_{27}^+ + 0.272727 x_{27}^- + 0.28 y_{14}^- + 0.3 y_{23}^+ \geq 1.$$

Feasible extreme point to this cut: $1(2,0)$, $2(2,2)$.

W.P.G. point is $1(1,0)$ and $2(0,2)$ with $\phi = 45$. This is hence current best. With $k = 45$ now, the second cut is:

$$0.6667 x_{17}^+ + 0.26087 x_{17}^- + 0.26087 x_{24}^+ + 0.291667 y_{13}^+ + 0.26087 y_{24}^- \geq 1$$

Feasible extreme point to the 2 cuts: $1(2,2)$ and $2(2,0)$.

W.P.G. point $\equiv 1(0,2)$ and $2(2,0)$ with $\phi = 50$. The 3rd cut is:

$$0.206897 x_{14}^+ + 0.4 x_{26}^- + 0.241379 y_{14}^- + 0.24 y_{26}^+ \geq 1.$$

Feasible extreme points to the 3 cuts: $1(2,0)$ and $2(2,2)$

W.P.G. point $\equiv 1(2,0)$, and $2(2,2)$ with $\phi = 80$. The 4th cut then is:

$$0.380952 x_{16}^- + 0.6667 x_{25}^- + 0.259259 y_{16}^+ + 0.24 y_{25}^- \geq 1.$$

Use of Algorithm FEAS to Determine an Extreme Point Feasible to the 4 Cuts Given Above

We first determine the maximum values (\bar{g}^t) which each cut expression on the left hand side can assume. This is a trivial procedure e.g.
 $\bar{g}^1 = 2(0.272727) + 1 + 2(0.28) + 2(0.3) = 2.705454$. Similarly,

$$\bar{g}^2 = 2.29348$$

$$\bar{g}^3 = 2.17655$$

$$\bar{g}^4 = 3.09376$$

$$\text{Also, } \bar{g} = \sum_{t=1}^S \left(\frac{1}{\bar{g}^t} \right) = 1.58832.$$

In our usual notation, clearly, $m_x = m_y = 3$. Letting the 3 distinct x coordinate locations $(0,1,2)$ be 1,2,3 respectively and the 3 distinct y locations $(0,1,2)$ be 1,2,3 respectively (where these are differentiated from the 3 for the x location by the value of the subscript k on λ_{ijk} as in Section 4.9), we have,

From Section 4.9.1:

Coefficients g_{ijk} of $\lambda_{ijk} = \sum_{t=1}^4 (g_{ijk}^t / \bar{g}^t)$ where, e.g. if $k = 1$, g_{ij1}^t is the contribution to cut # t of x_i , having $x_i = x_j$ ($j =$ one of the three distinct positions 1,2,3). Arranging in the "blocks":

Block #1 (For x_1)			Block #2 (For x_2)		
λ_{111}	λ_{121}	λ_{131}	λ_{211}	λ_{221}	λ_{231}
$g_{ijk} = 0.36$	0.319	0.62604	0.89934	0.51301	0.59711

Block #3 (For y_1)			Block #4 (For y_2)		
λ_{112}	λ_{122}	λ_{132}	λ_{212}	λ_{222}	λ_{232}
$g_{ijk} = 0.42879$	0.42537	0.42195	0.38264	0.41247	0.442307

By arranging these in ascending order in each block, we obtain the coefficients g_{ij} of γ_{ij} as: ($i \equiv$ refers to position; $j \equiv$ refers to block)
 $j = 1,2 \equiv x$ coordinates; $j = 3,4 \equiv y$ coords).

	Block #1 (x_1)			Block #2 (x_2)		
	γ_{11}	γ_{21}	γ_{31}	γ_{12}	γ_{22}	γ_{32}
$g_{ij} \equiv$	0.319	0.360	0.62604	0.51301	0.59711	0.89934

	Block #3 (y_1)			Block #4 (y_2)		
	γ_{13}	γ_{23}	γ_{33}	γ_{14}	γ_{24}	γ_{34}
$g_{ij} \equiv$	0.42195	0.42537	0.42879	0.38264	0.41247	0.442307

Ranked Extreme Points (Relative to ϕ_{P_E})	(x_1, y_1) POSN.	(x_2, y_2) POSN.	ϕ_{P_E}	g^1	g^2	g^3	g^4
(3,3,3,3)	(2,0)	(0,2)	2.396477	1.978181	0.66667	2.176552	1.3333
(3,3,2,3)	(2,1)	(0,2)	2.3930542	1.69818	1.95833	1.935173	1.59259
(3,3,1,3)	(2,2)	(0,2)	2.389633	1.41818	1.250	1.69379	1.851852

In the table above, the extreme points of problem P_E are ranked until $g^t \geq 1$ for $t = 1, 2, 3, 4$. The point (3,3,3,3) optimizes ϕ_{P_E} . The other 2 points above were "picked up" from the "adjacent points listed" below:

Solution Picked Up	Adjacent Points with $\phi_{P_E} \geq \bar{g} = 1.58832$	
1) (3,3,3,3)	(2,3,3,3) \rightarrow (2.130437) (3,2,3,3) \rightarrow (2.094247) (3,3,2,3) \rightarrow (2.3930542)* and (3,3,3,2) \rightarrow (2.36664)	
2) (3,3,2,3)	(2,3,2,3) \rightarrow (2.1270142) (3,2,2,3) \rightarrow (2.0908242) (3,3,1,3) \rightarrow (2.389633)* and (3,3,2,2) \rightarrow (2.3632172)	

Hence extreme point feasible to the 4 cuts is: 1(2,2) and 2(0,2).

W.P.G. point from this is: 1(2,2), 2(0,2) with $\phi = 70$. The 5th cut is:

$$0.32 x_{15}^- + 0.24 x_{24}^+ + 0.8 y_{15}^- + 0.352941 y_{24}^- \geq 1.$$

Extreme point feasible to the 5 cuts is 1(2,0) and 2(0,0).

W.S.G. point is 1(2,0), 2(0,0) with $\phi = 70$. The 6th cut based on this is:

$$0.32 x_{16}^- + 0.24 x_{23}^+ + 0.368421 y_{16}^+ + 0.66667 y_{23}^+ \geq 1.$$

Extreme point feasible to the 6 cuts is 1(2,1) and 2(0,1).

W.P.G. point is 1(2,1) and 2(0,1) with $\phi = 60$. The 7th cut then is:

$$0.347826 x_{16}^- + 0.26087 x_{23}^+ + 0.2667 y_{17}^+ + 0.571429 y_{17}^- + 0.6667 y_{27}^+ \\ + 0.294118 y_{27}^- \geq 1.$$

There is no extreme point feasible to the 7 cuts and we stop. The optimum location is 1(1,0) and 2(0,2); The optimum allocations are:

	1	2	3	4	5	6	7
1	0	0	10	0	0	15	10
2	0	0	0	20	5	0	0

The optimum objective function value is $\phi = 45$.

CHAPTER VI

COMPUTATIONAL RESULTS, SUMMARY AND CONCLUSIONS

6.1 Introduction

In this chapter, we will first report and compare computational times for the rectilinear location problem (with interactions) with those reported by Pritsker (33). We will then demonstrate the superiority of the Newton's search technique, which we specialized in Section 4.6 to help us in solving problems PAR 1 and PAR 2 , over Bolzano's search. Next, we will compare the times we obtained for the location-allocation problem with a single product and no interactions with those reported by Love and Morris (28) and by Sherali and Shetty (38).

After that, we will report computational times for our general multi-product, multi-source (with interactions) and multi-destination rectilinear location allocation problem. A multiple second order regression model will be developed to serve as a "prediction equation" and an analyses of variance will be carried out to examine the linear and quadratic effects of each factor in our problem. Certain simplifications and the limiting size problems will be then presented. (All computations will be carried out on a CDC Cyber 74 machine; execution time will be reported.)

We will finally summarize the particular contributions made in this dissertation and conclude by recommending some further research.

6.2 Computational Results

6.2.1 The Pure Location Problem (MFL0C)

Herewe have reproduced times reported by Pritsker (33) to compare

with those obtained by us. Pritsker has given conditions under which his solution procedure may not terminate at the true optimum. The entries in the last column headed "Percent of solutions declared optimal" are based on this.

For each combination of m and n , we generated five independent random problems which we solved four times each. The starting solution provided by the algorithm was used once, and randomly generated starting solutions were used for the remaining three times. The average times reported are based on the first starting solutions. The last column entries indicate that we obtained the same optimum solution for the case of randomly generated starting solutions, as for the original starting solution.

Table 1. Computation Experience with the Revised Pritsker-Ghare (PG) Facility Locating Procedure and with the MFLOC Procedure Presented in this Study (H)

Number of Existing Facilities m	Number of New Facilities n	Average Computational Time (Seconds)		% of Solutions Declared Optimal	
		PG*	H	PG*	H
10	4	0.1147	0.01082	100	100
20	5	0.1294	0.01189	90	100
100	20	1.3880	0.1380	95	100
400	80	28.5875	2.1815	0	100

*These times were obtained on a CDC 6500 computer.

6.2.2 Newton's Search vs. Bolzano's Search Technique

In Section 4.6, we had indicated how special advantage could be

taken of the piecewise linearity of the $\psi(\lambda_j)$ and $\bar{\psi}(\lambda_j)$ curves. The advantage which accrues from this is that we have to solve fewer transportation problems and yet arrive at a more accurate solution.

It may be mentioned in this context that the transportation code is used several times in the entire procedure and it is very essential to have an efficient transportation code to arrive at reasonable computational times. The one we used has been developed by Shetty et al. (39) and to our knowledge it is the fastest known procedure.

Table 2 shows the number of transportation problems which had to be solved to obtain a single $\bar{\lambda}_j$ for $k = 1$, $n = 2$ and $m = 7$ (in the usual notations) in the solution of problem PAR1. (For the pure Bolzano search (initiated at 9999999) and for the modified Bolzano technique used in (37), the procedure was terminated when $\psi(\bar{\lambda}_j)$ came within $\pm 5\%$ of the current optimum solution k . The accuracy figures relate this $\bar{\lambda}_j$ with that for which $\psi(\bar{\lambda}_j) = k$.)

Table 2. Newton's Search vs. Bolzano's Search Technique

Pure Bolzano Search		Modified Bolzano's Search Used in (37)		Specialized Newton's Search Used by Us	
Number of* Transportation Problems	Accuracy*	Number of* Transportation Problems	Accuracy*	Number of* Transportation Problems	Accuracy*
25	98.834%	9.5	98.87%	2.55	100%

*The figures reported are average figures obtained over 20 problems.

6.2.3 The Single Product Rectilinear Location Allocation Problem Without Interactions Between New Facilities

In this section, we will compare the times reported in references

(28) and (38) with those obtained by us for $K = 1$ and with no interactions between new facilities. (Love and Morris (28) solve these problems using a discrete space formulation. The superiority of the continuous space formulation in conjunction with discretization whenever convenient is evident from the Figures appearing in Table 3 below.)

Table 3. Comparison of Computational Experience with the Solution Procedures of Love and Morris, Sherali A., And the Proposed Procedure (i.e. (TLM), (TA), and (TH)).

m	n = 2			m	n = 3		
	TLM*	Restricted Capacity			TLM*	Restricted Capacity	
		TA**	TH			TA**	TH
	(secs)	(secs)	(secs)		(secs)	(secs)	(secs)
12	3.12	10.11	7.74	12	27.76	48.63	47.25
14	3.66	13.35	9.64	13	34.44	62.91	52.63
16	16.92	15.69	13.54	14	35.28	88.39	72.15
18	59.16	21.14	18.85	15	265.44	109.61	79.69
20	76.16	23.36	20.43	16	361.56	133.81	98.34
25	575.34	36.02	30.59	17	836.52	150.10	124.23
30	1992.78	43.30	42.47	18	1450.20	176.45	135.94
35	5479.92	54.67	53.66	19	2511.78	203.52	173.35

*These times are reported for the unrestricted capacity problems which have a much simplified technique for solution. Comparison of times obtained by Sherali and Shetty with these times are in reference (38).

** A CDC Cyber 74 Computer was used for this study (as in our study).

6.2.4 Computational Experience with the General Rectilinear Distance Location-Allocation Problem

We will now report computational times for our problem solution

procedure. We employ a 3^3 factorial design to report our results so that some meaningful analysis may be subsequently carried out.

Table 4. Computational Times for Problem GRLAP (seconds)

No. of New Facilities (n)	No. of Products (K)	Number of Existing Facilities (m)		
		9	12	15
2	1	4.67	7.74	10.99
3		13.84	47.25	79.69
4		49.23	93.70	149.01
2	2	7.35	26.02	26.56
3		15.51	38.35	83.67
4		94.99	162.07	270.18
2	3	19.95	39.98	61.58
3		67.43	177.98	341.22
4		212.87	419.25	790.09

6.2.5 Statistical Analyses for the Data of Problem GRLAP Computational Times

The mathematical model we will use to test significance of our factors N,M,K and their interactions is

$$T_{ijk} = \mu + N_i + M_j + K_k + NM_{ij} + NK_{ik} + MK_{jk} + NMK_{ijk}.$$

(where T is the computation time and μ is the overall mean.)

Due to the unequal number of problems run for each combination of N,M, and K, we averaged the observed computational times. As a result, we did not have any replications and hence no error term appears above. However, as is usually the practice in such cases, we will use our third order

interaction term NM_{ijk} as error to test the significance of the other factors.

We also decomposed the factors into their linear and quadratic effects (subscripted L and Q respectively in Table 5). Yate's Method for the 3^n factorial design analyses was used to arrive at the following results:

Table 5. 3^3 Factorial Analyses for Linear and Quadratic Effects

Source	Sum of Squares	Degrees of Freedom	Mean Squares	F-Statistic Ratios
N_i		2		
N_L	230421.38	1	230421.38	61.063***
N_Q	9503.4706	1	9503.4706	2.5185
M_j		2		
M_L	97849.883	1	97849.883	25.931***
M_Q	1391.9408	1	1391.9408	0.3689
NM_{ij}		4		
$N_L M_L$	51355.223	1	51355.223	13.6094**
$N_L M_Q$	1504.3538	1	1504.3538	0.39866
$N_Q M_L$	298.95564	1	298.95564	0.07922
$N_Q M_Q$	24.5884	1	24.5884	0.006516
K_k		2		
K_L	155726.46	1	155726.46	41.2683***
K_Q	23941.852	1	23941.852	6.34471*
NK_{ik}		4		
$N_L K_L$	88779.175	1	88779.175	23.5269***
$N_L K_Q$	11186.658	1	11186.658	2.96452
$N_Q K_L$	3148.4817	1	3148.4817	0.8344
$N_Q K_Q$	448.09111	1	448.09111	0.1187
MK_{jk}		4		
$M_L K_L$	43283.56	1	43283.56	11.47**
$M_L K_Q$	8084.2278	1	8084.2278	2.1424
$M_Q K_L$	1209.5325	1	1209.5325	0.32053
$M_Q K_Q$	176.29756	1	176.29756	0.04672
NMK_{ijk} (Error)	30188.127	8	3773.5158	
Total	758522.26	26		

***"Very highly significant" (At the 0.1 percent level).

**"Highly significant" (At the 1 percent level).

*"Significant" (At the 5% level).

We hence observe that all the three factors N , M and K have a strong linear effect. Also the $N_L K_L$ interaction i.e. the difference in times obtained by changing N linearly at different values of K is highly significant. Similarly, the $M_L K_L$ and the $N_L M_L$ effects are highly significant. The number of products has a significant quadratic effect. This indicates that with increasing problem sizes, the computational times will not tend to "blow-up" disproportionately because of the strong linear propensities.

A regression analyses run to fit a second order model in the ranges specified in Table 4 yielded the following prediction equation:

$$\text{COMPUTATION TIME} = 39.798 N^2 - 559.349 N + 1.692 M^2 - 121.497 M + 63.169 K^2 - 657.934 K + 21.806 NM + 86.013 NK + 20.019 MK + 1859.5913 \text{ (cpu secs)}.$$

6.2.6 Some Simplifying Approximations

In reference (37), it has been demonstrated how the cutting plane algorithm "hits" the optimum solution early in the procedure and almost 95% of the time is spent thereafter in exhausting the entire feasible region. Because of the increased depth of cut presented here, this would be all the more true. Indeed, the main advantage of this lies in the fact that the procedure may be terminated prematurely with a good deal of confidence that the current solution is the true optimum or is at least very close to it.

In Table 6 we report times for $K = 1$. The time reported is that taken to reach the solution which was current best when the procedure was terminated after 600 seconds of computation.

Table 7 gives the results for our general problem with $K = 5$. The entries have a similar significance as for $K = 1$ except that here, the run

was terminated after 800 seconds of computation.

Table 6. First Times to Best Recorded Solution (K=1; No Interactions)
(Time in Seconds) Between New Facilities

n	m = 50	m = 100
10	18.32	31.10
20	46.639	115.15

Table 7. First Times to Best Recorded Solution (K=5, With Interactions)
(Time in Seconds) Between New Facilities

n	m = 15	m = 20
5	39.42	58.15
10	72.46	139.5

To further support the hypothesis that Tables 6 and 7 contain times to reach the true optimum, the $10 \times 100 \times 1$ problem was run until the true global optimum was reached. The time required for this was 1279.19 seconds. The optimum value was indeed that which was reached in 31.10 seconds.

It may be noted that even in fairly large sized problems, if several existing facilities have the same x or y coordinates, the corresponding reduction in feasible extreme points yields considerable improvement.

6.3 Summary of Specific Contributions

By way of a summary, we will state the specific contributions made in this dissertation:

- 1) We developed a new algorithm to solve efficiently the interacting multifacility rectilinear distance location problem.
- 2) As a subset of the above problem, we provided a solution procedure for a graph theory problem which may be stated thus:

"Given a set of nodes with each node connected to every other node by an arc, and given weights associated with each node and with each arc, find a subset of these nodes such that the sum of the weights on the nodes and their associated arcs is maximized."

(The restriction of all nodes being interconnected may be relaxed by simply accommodating arcs of zero weight). The solution to this problem is precisely that which we used to effect joint movements of degenerate new facilities.

- 3) We developed a generalized extension of the cutting plane algorithm to generate deeper cuts than any other known procedure. This may be used in the context of any problem where cutting planes are employed.
- 4) As a step in the generation of the cut, we specialized Newton's Method to solve a sub-problem and hence made the procedure much more efficient and accurate than it was when using the conventional Bolzano search technique.
- 5) We developed an algorithm guaranteed to find a feasible integer point solution to the polyhedral set formed by the cutting planes. This algorithm may be extended to the general case where the feasible region is comprised of discrete points in space superimposed by a set of linear constraints.
- 6) Finally, we have developed an algorithm which can solve the rectilinear

distance location-allocation problem with multi-interacting new facilities, multi-destinations and several products and have analyzed its computational aspects.

6.4 Conclusions and Recommended Research

We have already spoken of the critical influence that the efficiency of the transportation code has on the computational times of our problem. It is our strong opinion therefore that any extra effort expended in accommodating the most efficient transportation code available is wisely spent.

Another point worth noting is that location-allocation problems as examined from the real-world point of view are, what are called, one-shot problems i.e. the only time they need to be solved is when designing the layout. Therefore, even if the problem is large sized and involves a fair amount of computer time, it is not prohibitively expensive. Besides, one can take advantage of the early attainment of the global optimum to compromise intelligently between risk and cost.

In the design of new facilities, often the capacity of the sources are also unknown quantities. In the light of this, a modification is possible where the problem may be run by assuming unlimited source capacities. The transportation problem for this case is trivially solved. The result is that not only does the computational time reduce considerably but we also get an indication of optimum facility capacity. In such a context, therefore, the unrestricted capacity problem is more realistic.

6.4.1 Some Recommended Research

In the solution techniques for the Bilinear Programming Problems,

Vaish (41) has introduced a method based on the inductive construction of a sequence of polytopes to arrive at the global optimum. A comparison study of this and our cutting plane technique may be carried out.

Another related study is to adapt the procedure presented here to other forms of distance measure, particularly the Euclidean distance measure. Geographically speaking, in large scale problems, this would seem to have meaningful applications.

In our presentation, we have considered demands to be deterministic. In several real world problems, demands are stochastic and often, it is possible to represent them by certain known probability density functions. A modification of this procedure to accommodate such a situation would be very interesting and useful.

Another factor which arises in real world problems is quantity discounts. Here the total cost of purchase is a monotonically increasing, piecewise linear concave function of the quantity purchased. Accounting for such possibilities would be an important and interesting feature.

BIBLIOGRAPHY

1. Balas, E., "Intersection-Cuts - A New Type of Cutting Planes for Integer Programming," Operations Research, Vol. 19, pp. 19-36 (1971).
2. Bazaraa, M. S. and J. J. Jarvis, "Linear Programming and Network Flows," 1975.
3. Bellmore, H. and H. D. Ratcliff, "Set Covering and Involutory Bases," Mgt. Science, Vol. 18, #3, 1971, pp. 194-206.
4. Bender and Goldman, "Optimization of Distribution Networks," NBS Report 6930 (1962).
5. Brenstall, Leaver and Sussands, "Evaluation of Transportation Costs for Alternate Factory Sites," Opns. Res. Qtrly. (13), 1962.
6. Burdet, C. A., "Polaroids - A New Tool in Nonconvex and in Integer Programming," Naval Research Logistics Quarterly, Vol. 20, 1973, pp. 13-22.
7. Cabot, V. A. and R. L. Francis, "Solving Certain Non-convex Quadratic Minimization Problems by Ranking the Extreme Points," Operations Research, Vol. 18, 1970, pp. 82-86.
8. Cabot, A. V., B. L. Francis and M. A. Stary, "A Network Flow Solution to a Rectilinear Distance Facility Location Problem," AIIE Transactions, Vol. 2, #2, 1970, pp. 132-141.
9. Cooper, L., "Location-Allocation Problems," Operations Research, Vol. 11, #1, 1963, pp. 331-343.
10. Cooper, L., "Solution to Generalized Locational Equilibrium Models," Journal of Regional Science, 7, 1967, pp. 1-18.
11. Cooper, L., "Heuristics for the Location-Allocation Problems," SIAM Review, 6, 1964, pp. 37-52.
12. Effroymsen, M. A. and T. L. Ray, "A Branch and Bound Algorithm for Plant Location," Operations Research, Vol. 14, #3, 1966, pp. 361-368.
13. Eyster, J. W., and J. A. White, and W. W. Wierwille, "On Solving Multifacility Location Problems using a Hyperboloid Approximation Procedure," AIIE Transactions, Vol. 5, #1, 1973, pp. 1-6.

14. Francis, R. L., "On Some Problems of Rectangular Warehouse Design and Layout," The Journal of I.E., Vol. 18, 1967.
15. Francis, R. L., and L. Richard, "Sufficient Conditions for Some Optimum Facility Design," Operations Research (15) 1967, pp. 448-466.
16. Hill, T. W. J. and A. Ravindran, "Technical Note on Programming with Absolute Value Functions," Journal of Optimization Theory and Applications, Vol. 17, # 1/2, 1975.
17. Hillier, F. S. and G. J. Lieberman, "Introduction to Operations Research," Holden-Day, 1972.
18. Ignizio, J. P., "A Heuristic Solution to the Generalized Covering Problem," Unpublished Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Va., 1971.
19. Khumawala, B. M. "An Efficient Branch and Bound Algorithm for the Warehouse Location Problem," Mgt. Science, Vol. 18, #12, 1972, pp. 718-731.
20. Kirby, M. J. L., M. R. Love and K. Swamp, "Extreme Point Mathematical Programming," Mgt. Science, Vol. 18, #9, May 1972, pp. 542-549.
21. Kuehn, A. A. and M. J. Hamburger, "A Heuristic Program for Locating Warehouses," Mgt. Science, Vol. 19, #4, 1963, pp. 643-666.
22. Kuenne, R. E. and R. M. Soland, "The Multi-Source Weber Problem: Exact Solutions by Branch and Bound," IDA Economic Papers (H. Williams, ed.), Program Analysis Division, Arlington, Va., 1971.
23. Kuhn, H. W., "On a Pair of Dual Non-Linear Problems," J. Abadie (ed.), Non Linear Programming, John Wiley, Interscience, N. Y. 1967.
24. Kuhn, H. W. and R. E. Kuenne, "An Efficient Algorithm for the Numerical Solutions of the Generalized Weber Problem in Spatial Economics," Journal of Regional Science, 4, 1962, pp. 21-34.
25. Lawler, E. L., "Covering Problems: Duality Relations and a New Method of Solution," Journal of the SIAM on Applied Mathematics, Vol. 14, #5, 1966, pp. 1115-1132.
26. Lehmer, "Teaching Combinatorial Tricks to a Computer," Proc. of Symposia in Applied Mathematics, Vol. X, Combinatorial Analyses, American Math. Society.
27. Love, R. F. and J. G. Morris, "Solving Unconstrained Location Problems Involving l.p. Distances using Convex Programming," Operations Research (U.S.), 23, #3, May-June 1975, pp. 581-587.

28. Love, R. F. and J. G. Morris, "Computational Procedure for Exact Solution of Location-Allocation Problems with Rectilinear Distances," Naval Research Logistics, September 1975.
29. Martos, B., "Subdefinite Matrices and Quadratic Forms," SIAM Journal of Applied Mathematics, Vol. 17, 1215-1223 (1969).
30. Martos, B., "The Direct Power of Adjacent Vertex Programming Methods," Mgt. Science, Vol. 17 (1965), pp. 241-252.
31. Morris, J. G., "Facilities Location and Location-Allocation Using Rectilinear Distances," Ph.D. Dissertation, University of Wisconsin, Graduate School of Business, 1972.
32. Murty, K., "Solving the Fixed-Charge Problem by Ranking the Extreme Points," Operations Research, Vol. 16, 1968, pp. 268-279.
33. Pritsker, A.A.B., "A Note to Correct the Procedure of Pritsker and Ghare for Locating New Facilities with Respect to Existing Facilities," AIIE Transactions, Vol. 5, #1, 1972, pp. 84-86.
34. Pritsker, A. A. B., and P. M. Ghare, "Locating New Facilities with Respect to Existing Facilities," AIIE Transactions, Vol. 2, #4, 1970, pp. 290-297.
35. Rao, M. R., "On the Direct Search Approach to the Rectilinear Facilities Problem," AIIE Transactions, Vol. 5, #3, 1973, pp. 256-264.
36. Revelle, C. and R. Swain, "Central Facilities Location," Geographical Analyses, Vol. 2, #1, 1970, pp. 30-42.
37. Sherali, A. D., "The Rectilinear Distance Location-Allocation Problem," Masters Thesis, Georgia Institute of Technology, 1976.
38. Sherali, A. D. and C. M. Shetty, "The Rectilinear Location-Allocation Problem," I. Sy.E. Report Series, J-76-11, Georgia Institute of Technology, April 1976.
39. Shetty, C. M., R. W. Langley, and J. Kennington, "Efficient Computational Devices for the Constrained Transportation Problem," Naval Research Logistics Quarterly, Vol. 21, #4, Dec. 1974, pp. 637-647.
40. Toregas, C. and C. Revelle, "Optimal Location Under Time or Distance Constraints," Paper of the Regional Science Association, Vol. 19, #6, 1971, pp. 1364-1373.
41. Vaish, H., "Nonconvex Programming with Applications to Production and Location Problems," Ph.D. Dissertation, Georgia Institute of Technology, 1974.

42. Wesolowsky, G. O., "Rectilinear Distance Location Under Minimax Optimization Criteria," Transportation Science (U.S.), 6, 1972, May, pp. 103-113.
43. Wesolowsky, G. O. and R. F. Love, "Location of Facilities with Rectilinear Distances Among Points and Area Destinations," Naval Research Logistics Quarterly (U.S.), 18, March 1971.
44. White, J. A., and R. L. Francis, "Facility Layout and Location," An Analytic Approach," Prentice Hall, 1974.